

News From The Summer Consumer Electronics Show

# COMPUTE!

\$2.95  
August  
1985  
Issue 63  
Vol. 7, No. 8  
\$3.75, Canada  
02195  
ISSN 0194-357X

The Leading Magazine Of Home, Educational, And Recreational Computing



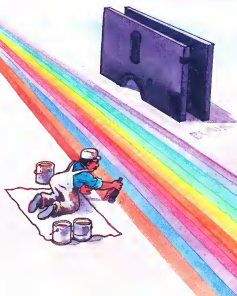
**Animator**  
The Easy Way To  
Create Your Own  
Colorful Graphics  
On IBM, Commodore,  
Apple, Atari, And TI

**Advanced  
Commodore 1541  
Disk Commands**  
Add Power  
To Your Programs

**Apple IIC Mousor**  
Efficient Editing  
With Fast  
Cursor Control

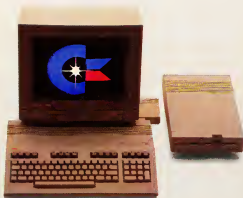
**Atari Color Mixing**  
How To Create  
Dazzling Color  
Displays

**IBM Filecopy**  
A Better Way  
To Make Backups





**If you own an Apple IIc,  
you'd have to add three more Apple IIc's,  
an Extra Keypad,  
30 Block Graphic Sets,  
Color Sprites,  
two more voices,  
four instruments,  
a Cartridge Port, a Joystick Port,  
and a Commodore 64...**



**to match the versatility, expandability  
and higher intelligence of the  
new Commodore 128**  
(and it costs less too).

The new Commodore 128™ personal computer is breakthrough technology at a breakthrough price. It outshines the Apple® IIc in performance capability, performance quality and price. It is expandable to 512K RAM. The IIc

doesn't expand. Commodore 128 has a numeric keypad built into its keyboard that makes crunching numbers a lot easier. And graphic and sound capabilities that far exceed those of the Apple IIc. But the most important news is that

Commodore 128 jumps you into a new world of business, productivity, education and word processing programs while still running over 3,000 programs designed for the Commodore 64.™ That's what we call a higher intelligence.

**COMMODORE 128  PERSONAL COMPUTER**  
A Higher Intelligence

\*Apple is a registered trademark of Apple Computer, Inc.

© Commodore 1985

# AMAZING DAISY

NOW! FULL SIZE, FULL FEATURE, LETTER QUALITY AT ONLY \$353

If you have been searching for a letter quality printer you have probably found the flood of claims and counterclaims to be a real roadblock in your search. Not long ago we were in the same position. We tried to determine which daisy wheel printer had all the features our customers wanted, yet would not set them back a month's salary. Recently several manufacturers have introduced machines that had features we were searching for. After a thorough assessment, we eliminated one model after the other for lack of one feature or another until we only had one left.

## THE RESULTS ARE IN

We found the printer which has all the features anyone could want. The winner is the Aproték Daisy 1120, a real heavy-duty workhorse printing at 20 characters per second. The manufacturer is Olympic Co., Ltd., a highly respected Japanese firm.

## FEATURES GALORE

This printer has it all. To start with, it has a front panel Pitch Selector button with indicators which allows 10, 12, 15 characters per inch (CPI) or Proportional Spacing. There is a Select (Online) button (with indicator) and a Line Feed button. You can also set Top-of-Form or Form Feed with the touch of the TOF button. Other front panel indicators include Power and Alarm.

To load a sheet of paper, simply place it in the feed slot and pull the paper bail lever. PRESTO! The paper feeds automatically to a 1 inch top margin and the carriage aligns to the selected left margin. In this manner, each page can have identical margins automatically. You can continue to compute while the Daisy 1120 is

printing. The built in 2K buffer frees up your computer while printing a page or two allowing you to go to your next job.

To really put your printer to work, the Cut Sheet Feeder option is great for automatic printing of those long jobs. Also available is the adjustable Tractor Feed option. Compare our option prices! Best of all the Daisy 1120 is quiet: only 57 dB-A (compare with an average of 62-65 dB-A for others).

## COMPLETE COMPATIBILITY

The Daisy 1120 uses industry standard Diablo® compatible printwheels. Scores of typeface styles are available at most computer or stationery stores. You can pop in a 10, 12, 15 pitch or proportional printwheel and use paper as wide as 14". At 15 CPI you can print 165 columns—great for spreadsheets.

The Daisy 1120 uses the Diablo Hytype II® standard ribbon cartridges. Again universally available.

Not only is the hardware completely compatible, the control codes recognized by the Daisy 1120 are Diablo 630® compatible (industry standard). You can take advantage of all the great features of word processing packages like Wordstar®, pfs: Write®, Microsoft Word® and most others which allow you to automatically use superscripts, subscripts, automatic underlining, boldface (shadow printing) and doublestrike.

The printer has a set of rear switches which allow the use of standard ASCII as well as foreign character printwheels. Page length can be set to 8, 11, 12, or 15". The Daisy 1120 can also be switched to add automatic line feed if required.

## THE BEST PART

When shopping for a daisy wheel printer with all these features (if you could find one), you could expect to pay \$600 or \$700 dollars. The options would add much more. Not now! We have done our homework. We can now offer this printer for only \$353. Order yours today!

## NO RISK OFFER

Try the Daisy 1120 for 2 weeks. If you are not satisfied for ANY reason we will refund the full price—promptly. A full 1-year parts and labor warranty is included.

## THE BOTTOM LINE

Aproték Daisy 1120 (Order#1120) \$353 w/standard Centronics parallel interface and 2K buffer.

## Options

Auto Cut Sheet Feeder (#1110) \$188

Tractor Feed (#1112) \$77

## Accessories

8' Cable for IBM PC® and compatibles (#1103) \$26

Interface with cable: •TI-99/4A (#106) \$66

•Apple II or IIe (#1104) \$76

•All Commodore (except Pet) (#1105) \$44

•All Atari (#1107) \$66

Shipping is \$11—UPS continental USA. If you are in a hurry, UPS Blue or Air Paced Post (second day air) is \$25. Canada, Alaska, Mexico and Hawaii are \$30 (air). Other foreign is \$60 (air). California residents add 6% tax. Prices are cash prices—VISA and MC add 3% to total. We ship promptly on money orders, cashier's checks, and charge cards. Allow 14-day clearing for checks. No C.O.D.'s. Payment in US dollars only.

TO ORDER ONLY CALL TOLL FREE

(800) 962-5800 USA

(800) 962-3800 CALIF. (8-5 PST)

Or send payment to address below:

Technical Information & Customer

Service: (805) 987-2454 (8-5 PST)

Dealer Inquiries Invited

©1985 APOTÉK. All rights reserved.  
Trademark: Diablo, Hytype II, 630-Series  
Corp., Wordstar-Micropro Corp., PFS-  
Software Publishing Corp., Microsoft  
Word-Microsoft Corp., Apple II,  
De Apple Computer, Inc.  
IBM PC-IBM Corp.,  
PST 68K



**APOTÉK**

1071-A Avenida Acero, Camarillo, CA 93609

# COMPUTE!

AUGUST 1985  
VOLUME 7  
NUMBER 8  
ISSUE 63

## FEATURES

- 14 Report from the Summer Consumer Electronics Show ..... Tom R. Halithill  
16 Monster Memory .....  
20 A tantalizing Peek at the Amiga .....  
42 Animator ..... Steve Johnson

## GUIDE TO ARTICLES AND PROGRAMS

AP/AT/64/12B  
TI/PC/PCjr

## REVIEWS

- 38 Archon II: Adept ..... Arthur Leyenberger  
38 WordPerfect for IBM ..... Richard Mansfield  
40 Adventures in Namia for Apple and 64 ..... C. Regena

AT/64/12B/AP  
PC  
AP/64

## COLUMNS AND DEPARTMENTS

- 4 The Editor's Notes ..... Robert Lock  
8 Readers' Feedback ..... The Editors and Readers of COMPUTE!  
24 The Beginner's Page ..... Tom R. Halithill  
26 Computers and Society:  
Compilers, Interpreters, and Flow, Part 2 ..... David D. Thornburg  
28 On the Road with Fred D'ignazio:  
Buying the Right Educational Software ..... Fred D'ignazio  
32 HOTWARE .....  
33 Telecomputing Today—SIGs: Behind the Scenes ..... Arlan R. Levitan  
34 INSIGHT: Atari—Atari Input/Output ..... Bill Wilkinson  
35 IBM Personal Computing: New Life for Aging PCs ..... Donald B. Trivette  
36 Programming the TI: Trivia Quiz ..... C. Regena

AT  
PC/PCjr  
TI

## THE JOURNAL

- 62 Archive: Two-Drive Backup for Commodore 64 ..... Philip I. Nelson  
68 Atari Color Mixing ..... Karl E. Wiegiers  
71 Mouser:  
Escape Mode Cursor for the Apple IIc ..... J. Blake Lambert and Tim Victor  
72 Commodore 64 Headliner ..... Robert F. Lambiase  
74 Using the Commodore USB Function ..... Keith R. Bergerstock  
76 Sound and Music on the Commodore 128, Part 1 ..... Philip I. Nelson  
78 Colorful Text for IBM Graphics ..... Peter F. Nicholson, Jr.  
80 Advanced 1541 Disk Commands ..... Dave Straub  
82 IBM Filecopy ..... John Klein and Jeff Klein  
84 Apple Text Windows ..... Daniel L. Joynt

64  
AT  
  
AP  
64  
V/64/+4/16/12B  
12B  
PC/PCjr  
V/64/+4/16/12B  
PC/PCjr  
AP

- 37 CAPUTE! Modifications or Corrections to  
Previous Articles  
41 Classified  
86 COMPUTE!'s Guide to Typing in Programs  
89 MLX Machine Language Entry Program for  
Commodore 64  
96 Advertisers Index  
96 Product Mart

NOTE: See page 85  
before typing in  
programs.

TOLL FREE Subscription Order Line 800-334-0868 (In NC 919-275-9809)

AP Apple, Mac Macintosh,  
AT Atari, V VIC-20, 64 Commodore  
64, +4 Commodore  
Plus/4, 16 Commodore 16,  
128 Commodore 128, P  
PET/CBM, TI Texas Instru-  
ments, PC IBM PC, PCjr IBM  
PCjr, CC Radio Shack Color  
Computer  
\*General interest

## COMPUTE! Publications, Inc.

One of the ABC Publishing Companies:  
ABC Publishing, President, Robert G. Burton  
1330 Avenue of the Americas, New York, New York 10019

Address all inquiries to:  
P.O. Box 5406, Greensboro, NC 27403

COMPUTE! The Journal for Progressive Computing (USPS 837250) is published monthly by COMPUTE! Publications, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone (919) 275-9809. Editorial Offices are located at 324 West Wendover Avenue, Greensboro, NC 27408. Domestic Subscriptions: 12 issues, \$24. Send subscription orders or change of address (P.O. form 3579) to COMPUTE! Magazine, P.O. Box 10954, Des Moines, IA 50342. Second class postage paid at Greensboro, NC 27403 and additional mailing offices. Entire contents copyright ©1985 by COMPUTE! Publications, Inc. All rights reserved. ISSN 0194-357X.

# Editors Notes

The subdued pallor of the personal computer section at the Summer Consumer Electronics Show was somewhat sobering. Dozens of industry vendors simply chose not to exhibit; dozens more have disappeared in the months since the last show. Noticeable in the reduced clutter of exhibitors was the increased level of professionalism and sophistication of presentation among those present. Also noticeable was the lack of industry-shaking innovation we've grown accustomed to over the last few years. Among the bright spots were our old friends at Atari, the Tramiels. They highlighted that which is best among us by promising new innovations and continued leadership at the cutting edge of truly consumer-oriented electronics. Their demonstration of an Atari/compact disc interface which allows an entire multi-volume encyclopedia to be stored and quickly retrieved from less than one-quarter of a single compact disc is truly significant. Their proposed pricing for new Atari ST systems promises hope for fall. (See the Consumer Electronics Show article elsewhere in this issue for more information.)

Commodorians are properly pushing the 128 system and reluctantly admitting the coming of the Amiga. We were shocked to discover that apparently some at Commodore still enjoy political magazine games.

Several of our competitors had already received Amiga systems while Commodore public relations personnel were currently telling us that all magazines would be treated equally. It makes one wonder what motive Commodore might have for withholding access to the Amiga from the largest Commodore-related publisher in the industry. Ah, well. COMPUTE! always perseveres, and you may rely on us to bring you continuing and timely assessments of the new Amiga. Among our articles this month on the Consumer Electronics Show, you'll find some early information on the Amiga. It looks like a pretty impressive machine.

On this increasingly hopeful note, we'll point out that the traditionally upbeat Christmas season, while viewed with caution, is expected to be a good one for the vendors who have remained in the marketplace. It's a bit of the smaller pie and fewer slices phenomena. That same principle can perhaps be extended to the magazine publishing industry. We have a small group among our competitors whose attacks on us over the years have ebbed and flowed with the success of the

various magazines they launch to compete with ours. As problems arise for whatever flagship they're currently pushing, we can detect a significant increase in the various voices they raise in criticism of us, our style, our policies, our editors, our writing. We have always chosen to remain silent in the face of these rumblings and time has always proven to be our steadfast ally. We suspect such will remain the case. In the meantime, we'll continue our efforts to always provide you with the most balanced magazines of the best quality we can publish. Thank you for your continued support.



Editor in Chief



# THE BETTER LETTER BOX

**Introducing EasyPlex™.** The new, easy-to-use electronic mail system from CompuServe.

Finally! Electronic Mail that's so easy to use you can start composing and sending messages the first time you get online.

Designed for various experience levels, EasyPlex has a menu mode with simple, easy-to-follow directions for beginners, and it lets experienced users save time by working in the prompt or command modes. With EasyPlex, you can compose, edit, send, file, and take advantage of sophisticated

options previously available only with more expensive services.

CompuServe's EasyPlex lets friends and relatives, associations and club members communicate any time of the day or night. And small business owners, real estate professionals, insurance agents, lawyers, writers, etc. can communicate quickly and simply—either interoffice or interstate. "It's Easy." "Just Plex it!"

Best of all, EasyPlex is available to all CompuServe subscribers. And, along with EasyPlex, you get hundreds of valuable and entertaining computing options. Plus the assurance of belonging to the largest, fastest

growing computer information service in the world and the premier supplier of business information to FORTUNE 500 companies.

Start communicating! To buy a CompuServe Subscription Kit, see your nearest computer dealer. To receive our informative brochure or to order direct, call or write:

**CompuServe®**

Information Services, P.O. Box 20212  
5000 Arlington Centre Blvd., Columbus, Ohio 43220

**800-848-8199**  
In Ohio, call 614-457-0802

An H&R Block Company

**Publisher**  
Editor in Chief  
Director of Administration

**Senior Editor**  
Managing Editor

**Editor**  
Assistant Editor

**Production Director**  
Production Editor

**Editor, COMPUTE's GAZETTE**  
Technical Editor

**Assistant Technical Editor**  
Program Editor

**Features Editor**  
Assistant Editor, COMPUTE's GAZETTE

**Features Editor**  
Research Assistant

**Programming Supervisor**  
Editorial Programmer

**Submissions Reviewer**  
Programming Assistant

**Copy Editor**  
Executive Assistant

**Administrative Assistant**  
Associate Editors

**Contributing Editor**

**COMPUTE's Book Division**  
Editor

**Assistant Editor**  
Administrative Assistant

**Director, Book Sales & Marketing**  
Assistant

**Production Manager**  
Art & Design Director

**Assistant Editor, Art & Design**  
Mechanical Art Supervisor

**Artists**  
Typesetting

**Illustrator**

**Director of Advertising Sales**  
Production Coordinator

**Administrative Assistant**

**Promotions Assistant**

**Customer Service Manager**  
Desktop Sales Supervisor

**Assistant**

**Individual Order Supervisor**  
Assistant

**Warehouse Manager**  
Staff

**Data Processing Manager**  
Assistant

**Vice President, Finance & Planning**  
Executive, Finance & Planning

**Accountant**  
Financial Analyst

**Staff**

**Credit Manager**  
Staff

**Purchasing Manager**

**Stephen Levy**  
Gregg Kessel, J. Blake Lambert  
Laura Maggadden

**Steve Mayotte**  
Carol Dickerson

**Ima Swan**  
James R. Fory

**Lee Noel**  
Debbie Wray, Doreen Glick

**Terry Cash**  
Carole Dutton  
Harry Blair

**Ken Woodard**  
Paul Stokes  
Kathleen Stanton

**Caroline Dale**

**Philippa King**  
Gail Jones  
Debi Galtzoff, Jo Kusanen-Jones

**Rhonda Savage**  
Judy Taylor  
Betty Allen, Gayle Benbow,  
Mary Hunt, Jenna Nash, Chris Perry

**Lorrie Jordan**  
Harold Ayres, Steve Bowman,  
Lory O'Connor, David Hensley

**Leon Stokes**  
Chris Cox

**Paul J. Meglola**  
R. Steven Welter  
Robert J. Reel  
Karen K. Rogstad

**Dore Branch**  
Jill Pope

**Dore L. Beck**  
Byrl A. Gipe, Anne Ferguson, Pat  
Fulter, Dore Hall, Linda Miller,  
Mary Wadsworth, Jane Wraga

**Greg L. Smith**

**Robert C. Lock, Chief Executive Officer**  
Gary R. Ingvald, President

**Paul J. Meglola, Vice President, Finance and Planning**  
Debi Nash, Executive Assistant

**Arifa Arnefeld, Assistant**

**COMPUTE Publications, Inc. publishes:**

**COMPUTE's**  
COMPUTE's BOOKS  
COMPUTE's DISK

**Corporate Office:**  
324 West Wendover Avenue  
Suite 200  
Greensboro, NC 27601 USA

**Mailing address: COMPUTE's**  
Post Office Box 914  
Farmingdale, NY 11737

**Telephone: 919-275-9889**

**Coming in Future Issues**

**Hands-On Reports:**  
Atari 520ST  
Commodore Amiga  
Atari 130XE And DOS 2.5

**Word Search:**  
Puzzle Generator For Atari,  
Commodore 64, 128, VIC,  
Plus/4, 16, Apple,  
IBM PC/PCjr, TI-99/4A

**Chess For IBM PC/PCjr**

**Easy Apple Screen Editing**

**Commodore 64 Disk Commander**

**The Last Warrior**  
Action Game For Commodore 64,  
Atari, Apple, IBM PC/PCjr

**Subscription Orders**

**COMPUTE Circulation Dept.**  
P.O. Box 914  
Farmingdale, NY 11737

**TOLL FREE Subscription Order Line**  
**800-334-0868**  
In NC 919-275-9889

**COMPUTE Subscription Rates**  
(12 Issue Year):

**US** (one yr.) \$24  
(two yrs.) \$45  
(three yrs.) \$65

**Canada and Foreign**  
Surface Mail \$30  
Foreign Air  
Delivery \$65

**Advertising Sales**

**1. New England**  
Jonathan M. Just  
Regional Manager  
212-315-1665

**2. Mid Atlantic**  
John Sovol  
Eastern Advertising  
Manager  
212-315-1665

**Kathy Hicks**  
Monisha A. Giffelman  
215-646-5700  
Brian S. Rogers  
212-674-0238

**3. Southeast & Foreign**  
Harry Blair  
919-275-9809

**4. Midwest**  
Gordon Benson  
312-362-1821

**5. Northwest/Texas**  
Phoebe Thompson  
408-354-5553

**6. Southwest**  
Ed Winchell  
213-378-8361

**Director of Advertising Sales**  
Ken Woodard

**COMPUTE Home Office 919-275-9889.**

**Address all advertising materials to:**  
Paul W. Stokes  
Advertising Production Coordinator  
**COMPUTE Magazine**  
324 West Wendover Avenue,  
Greensboro, NC 27608

**Robert C. Lock, Chief Executive Officer**  
Gary R. Ingvald, President

**Paul J. Meglola, Vice President, Finance and Planning**  
Debi Nash, Executive Assistant

**Arifa Arnefeld, Assistant**

**Member**  
**ABC**  
**Advert Bureau of Circulation**

**MPA**  
Magazine Publishers Association

**REI, IBM, VIC-20 and Commodore 64 are trademarks of Commodore Business Machines, Inc. and/or Commodore Electronics Limited.**  
Apple is a trademark of Apple Computer Company.  
IBM PC and PCjr are trademarks of International Business Machines Inc.

**ATARI is a trademark of Atari, Inc.**  
TI-99/4A is a trademark of Texas Instruments, Inc.  
Radio Shack Color Computer is a trademark of Tandy Inc.



# A SPECIAL OFFER FROM COMPUTE! BOOKS

Buy any of the new Apple titles listed below before September 15, 1985 and receive a \$1.00 discount on each book. These information-packed books contain games, applications programs, buying guides, step-by-step instructions, product reviews, special tutorials for kids, and much more for your Apple II, II+, IIc, IIfx, or Macintosh computer.



## COMPUTE! Kids and the Apple

Edward H. Catton  
\$12.95  
A fun and easy-to-use book for kids ages 10-14 interested in learning BASIC programming on their Apple computers.

## COMPUTE! First Book of Apple

Edited  
\$12.95  
A collection of 35 exciting games, educational programs, home applications, and graphics routines for owners of Apple II, II+, IIfx, and IIc computers.

## COMPUTE! Easy BASIC Programs for the Apple

Brian Flynn  
\$14.95  
A wide-ranging selection of short BASIC programs for the Apple II series of computers, including thought-provoking games, home applications, and business forecasting tools.

## COMPUTE! Guide to Telecomputing on the Apple IIc

Thomas E. Enright, Joan Nickerson, and Anne Wayman  
\$9.95  
An informative and easy-to-understand book for the beginner which explains how to access bulletin boards, select telecommunications software, and use the Apple IIc to communicate with other computers over the phone.

## COMPUTE! Apple Games for Kids

Clark and Kathy H. Kold  
\$12.95  
An instant library of quality educational software to teach and entertain the entire family.

## Becoming a MacArtist

Vahid Gushkiyan  
\$17.95  
An excellent illustrated guide to Macintosh graphics, giving you direct access to the power and flexibility of MacDraw, MacPaint, and MacWrite.

Mail in the coupon below with your payment by September 15, 1985 to receive your special discount from COMPUTE! Books.

To order your Apple titles, call toll-free 800-334-0868 (In NC 919-275-9809), or mail this coupon with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403.

— COMPUTE! First Book of Apple, 69-8 \$12.95  
— COMPUTE! Kids and the Apple, 76-0 \$12.95  
— Becoming a MacArtist, 80-9 \$17.95  
— COMPUTE! Easy BASIC Programs for the Apple, 85-4 \$14.95  
— COMPUTE! Apple Games for Kids, 91-4 \$12.95  
— COMPUTE! Guide to Telecomputing on the Apple IIc, 98-1 \$9.95

Subtotal \$ \_\_\_\_\_

Less \$1.00 per book discount \_\_\_\_\_

NC residents add 4.5% sales tax \_\_\_\_\_

Add \$2.00 PER BOOK for shipping \_\_\_\_\_

Total payment enclosed \_\_\_\_\_

Please allow 4-6 weeks for delivery. All payments must be in U.S. funds.

### ALL ORDERS MUST BE PREPAID

☐ Payment enclosed (check or money order)  
☐ Charge ☐ Visa ☐ MasterCard ☐ American Express

Account No. \_\_\_\_\_ Exp. Date \_\_\_\_\_ (if required)

Signature \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_

State \_\_\_\_\_ Zip Code \_\_\_\_\_



# Readers Feedback

The Editors and Readers of COMPUTE!

If you have any questions, comments, or suggestions you would like to see addressed in this column, write to "Readers Feedback," COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. Due to the volume of mail we receive, we regret that we cannot provide personal answers to technical questions.

## Modular Phone Booths

I'm writing a book—not on computers, but on radio news. Like you, we make great use of the telephone for relaying material recorded on cassette. Ours is analog voice material, and the similarity of what we do to computer applications of the phone prompted me to write.

I've tried without success to interest Ma Bell and GTE in installing modular jacks on their pay phones. This would allow us to use a simple patch cord to go from a cassette recorder output without wrenching off the phone handset cover or using an acoustic coupler, which lowers quality.

Are you aware of any attempts by computer users (or manufacturers) to get direct access to phone equipment? I would imagine this would be valuable for both groups, doing away with the need for acoustic couplers, plus their extra cost and size.

I'd appreciate hearing of any efforts you're aware of on pay phone access. It may take the clout of manufacturers, computer users, and computer publications to convince these giant phone utilities to allow direct pay phone access.

F. Gifford

We haven't heard of any such lobbying efforts among computer hobbyists, but your most likely allies would be user groups that cater to portable computer owners. For instance, there's a special interest group (SIG) on the CompuServe Information Service for users of the Radio Shack TRS-80 Model 100. As active members of a commercial telecomputing network, these people are also likely to have encountered the same problems that you have. Battery-powered lap portables with built-in direct-connect modems are wonderfully convenient for traveling journal-

ists and business people, but as you point out, the acoustic cups necessary to link them to pay phones are bulky, clumsy, and less reliable.

However, it seems unlikely that the phone companies will bend to your demands anytime soon. For one thing, portable computer users (and radio journalists) encompass a pretty small minority at this time—too small, we suspect, to justify modifying all the pay phones in the country. More importantly, handsets attached to public phones with modular jacks would be tempting targets for thieves. Anybody could unplug the handset and run off with it. Of course, pay phones could be redesigned with a conventionally attached handset and a modular jack as an accessory. Perhaps this will happen someday when portable computers are built into wristwatches and nearly everybody has one.

By the way, while you're lobbying, you might also want to target hotels and motels—we've found that many of them don't equip their phones with modular jacks, either.

## Fate Of The PCjr

Being the owner of a PCjr and with the recent bad news from IBM, it seems I have to make a decision on my future with the Junior. Hopefully you can give me some insight.

1. I could sell it and then buy a PC, but that would cost a thousand or two more for a system with similar color capabilities.

2. I could move to an Apple IIc, but I would have to start all over with my software.

3. I could make the Junior as PC-compatible as possible.

I would like more help with this third choice. I have heard of two expansion chassis, one by Quadram and another by Racore. Both add a second floppy drive, clock, parallel printer port, etc. And they add a switch to change modes from PCjr to PC. The Racore also adds an optional ten-megabyte hard disk.

Could you test these add-ons? Which is better, a second floppy drive or a hard disk? Will these chassis help

to secure what I've invested in the Junior, or should I bail out altogether?

Bob Hana

There's no reason to get rid of your PCjr as long as it meets your needs—and that's something only you can decide. IBM has not abandoned the PCjr; although production has been halted, IBM promises to continue supporting the computer with service and software. Since the PCjr already is fairly compatible with the PC, a wide selection of software is available and will continue to be available.

According to estimates we've seen, roughly 300,000 PCjrs have been sold. That's not a huge base compared to Commodore, TI, Apple, and Atari computers, but it's large enough to guarantee that software and expansion hardware will remain in supply in the immediate future. Still, in time, PCjr-specific products—particularly from non-IBM suppliers—may begin to dry up. So if there's anything you think your system might need, you should plan to buy while it remains available.

If you need to make your PCjr more PC-compatible, you must balance the cost of expanding the Junior against the cost of a new PC or compatible. There are several expansion modules on the market in addition to the products you mention which add more RAM, a second floppy disk drive, a real-time clock, parallel printer port, hard disk drive, and so on. Some of them allow more expansion than others and different combinations of options. See the September 1984 issue of COMPUTE! PC & PCjr magazine for reviews of the Tecmar JrCaptain and Legacy expansion modules.

Be aware, however, that no matter which one you pick, your PCjr won't be 100 percent PC-compatible 100 percent of the time because of some fundamental design differences. (See "PCjr Memory Compatibility," COMPUTE! March 1985.) Usually this isn't a major concern, but you should test new software on the PCjr before buying, or at least secure return privileges in case the program doesn't work.

The question of whether a hard disk is preferable to a second floppy drive depends on your needs and your pocketbook. A hard disk is much faster and stores much more data than a floppy drive, but it

# WHEN BATTERIES INCLUDED SET OUT TO DESIGN THE VERY BEST WORD PROCESSOR FOR ATARI® COMPUTERS... THEY FOUND THEY ALREADY HAD IT.

For  
COMMODORE  
and ATARI!



**BATTERIES INCLUDED**

**PAPER@LIP**

**Professional Word-Processor**

**Features all standard word processing functions PLUS:**

- Full Screen Editing
- Dual text windows with cut and paste between windows.
- Video preview with scrolling, lets you see exactly what will be printed, before it is printed.
- Easy text between documents — easily.
- Horizontal scrolling.
- Create wide documents with ease.
- Re-arrange words, phrases, sentences, paragraphs, and columns with just a few keystrokes.
- More than 40 different printer drivers — plus printer editor allows programming for any conceivable new printer.
- Easy to learn, easy to use.

**COMPATIBLE WITH ATARI 400, 800, 600 XL, 800 XL, 1200 XL, 65 XE, 130 XE**

**PAPER@LIP**

Few word processors have allowed Atari users to tap the full resources of their computer until Atari Paper Clip...

Atari Paper Clip is an extremely powerful, fully featured word processor that will allow your Atari to operate to the limits of its potential, with an ease of operation and speed you've never thought possible.

**PAPERCLIP FOR ATARI® AND COMMODORE® OWNERS  
WHO WANT THE VERY BEST IN WORD PROCESSING.**

**BATTERIES**



**INCLUDED**

**"The Energized Software Company!"**

WRITE TO US FOR FULL COLOUR CATALOGUE of our products for COMMODORE, ATARI, APPLE and IBM SYSTEMS  
FOR TECHNICAL SUPPORT OR PRODUCT INFORMATION PLEASE PHONE (416) 881-9816

30 Mural Street  
Richmond Hill, Ontario  
L4B 1B5 CANADA  
(416) 881-9941  
Telex: 06-21-8290

17875 Sky Park North, Suite P  
Irvine, California  
USA 92714  
(416) 881-9816  
Telex: 509-139

costs a lot more, too. One thing to keep in mind is that some commercial software is copy-protected in such a way that it requires you to boot off the floppy even if you have a hard disk.

## Commodore INPUT Revisited

Your answer to Scott Mefferd's letter about suppressing the Commodore INPUT question mark (COMPUTE, May 1985) is incorrect. It's quite easy to suppress the question mark that INPUT usually prints. Use POKE 19,64 before the INPUT command to disable it, and POKE 19,0 to bring it back. Here is an example:

```
10 POKE 19,64:INPUT"ENTER  
WORD":AS  
20 POKE 19,0:PRINT
```

You must enter some value when using this method (you can't just press RETURN). Add a PRINT statement after the input, since the cursor doesn't automatically go to the next line. You can also treat the keyboard as a peripheral, reading it with an INPUT# statement as shown here:

```
10 OPEN 1: PRINT"ENTER  
WORD":INPUT#1,AS  
20 PRINT:CLOSE 1
```

David Tucci

A number of readers have written to suggest these methods, both of which work fine. The first method is simple and trouble-free provided you always restore things to normal with POKE 19,0. The second method takes advantage of the fact that the keyboard is just another peripheral (device number 0) as far as the computer is concerned. You can OPEN a communication channel to the keyboard and input a string with INPUT#, the same as with other peripherals.

A third method, suggested by reader Robert Kodadek, bypasses the BASIC INPUT routine and calls CHRIN directly. CHRIN is a machine language routine stored in the computer's Read Only Memory (ROM) which fetches one character from the designated input device every time it is called. Since the keyboard is the computer's default input device (unless you specify otherwise), CHRIN acts much like GET, retrieving one character at a time:

```
10 AS="" : PRINT "ENTER WORD":  
20 SYS 65487. A=PEEK(780): IF  
A<13 THEN AS=AS +  
CHR$(A):GOTO 20  
30 PRINT:PRINT AS
```

CHRIN stores the character's ASCII value in the microprocessor's accumulator register, which is echoed at location 780 in the Commodore 64 and VIC-20. If you have a Plus/4 or Commodore 16, substitute the address 2034 for 780 in line 20. This method is a little slower than the first

two because it has to compile the string one character at a time in BASIC, terminating when it detects a carriage return (CHR\$(13)).

## Missing Atari Memory?

When I run the memory test on my Atari 800XL, it seems to check only the first 40K of RAM. There are no red blocks anywhere on the screen, but it refuses to check the last 8K of user RAM. When I check RAMTOP with PEEK(106), it returns a value of 160. If I am not mistaken, 48K of RAM should return a value of 192. The only other symptom is an above-average amount of keyboard lockup. What's wrong here?

Dave Nessell

Either you did not disable BASIC on powerup by holding down the OPTION key or you have a cartridge installed. A cartridge or the built-in BASIC uses the top 8K of your 48K of memory. To free up this 8K of RAM, disable BASIC or remove the cartridge when running the memory test.

The keyboard lockups are probably unrelated to the results of the memory test. Instead, BASIC is most likely to blame. The first Atari BASIC cartridge suffered from a lockup bug that was supposedly fixed in revision B BASIC, the version built into the 600XL and 800XL. Unfortunately, the fix only made the problem worse. (See "INSIGHT: Atari," COMPUTE, May and June 1985.)

Atari has finally eliminated the lockup bug for good in revision C BASIC. This version is built into the new 130XE computer and is available on cartridge for earlier machines. To obtain a cartridge, send \$15 to:

Atari Corp.  
Customer Relations  
390 Caribbean Drive  
Sunnyvale, CA 94088

## Resetting The SID Chip

Does SYS 64738 completely reset the Commodore 64 to its power-up state? When I use this SYS after running a music program, and then run a game program, I can hear a faint lingering tone. This does not happen when I turn the computer off and on, then run the game program.

Bruce Snider

You've noticed a 64 "feature" that many programmers overlook. Though you might expect system reset to clear the 64's SID (Sound Interface Device) chip, all it does is turn the volume down. This is easy to demonstrate. Turn up the volume on your TV or monitor and enter the following line in direct mode (without a line number):

POKE 54273,20:POKE 54277,15:POKE

54278,240:POKE 54276,33:POKE 54296,15

Press RETURN after typing this line.

The SID chip produces a continuous tone. Now type SYS 64738 and press RETURN, or press RUN/STOP-RESTORE. The volume cuts off (you may still hear a faint tone in the background). Enter POKE 54296,15 to turn up the volume again, and the tone comes back loud and clear, proving that the other SID registers retained the values you POKED in.

SYS 64738 makes the computer jump into ROM and execute several reset routines. One of these ROM routines—called IOINIT—is supposed to reset the system for normal input/output operations (IOINIT also executes when you press RUN/STOP-RESTORE). Unfortunately, rather than putting zeros in all 25 of the SID chip's control registers (as it should do to turn off the whole chip), IOINIT just puts a zero in the volume register (location 54296). If other SID registers are still active, crosstalk signals may leak through to the chip's output wire, producing background noise even though the SID's volume is off.

Besides adding unwanted crosstalk, residual SID values can prevent sounds from being heard. The three low bits of location 54295 control whether any of the SID's three voices are routed through the SID filter. If any of these three bits are left on (set to 1) and the filter cutoff frequency remains at an extreme value, one or more of the voices may be distorted or inaudible.

To eliminate such problems, use the statement FOR J=54272 TO 54296: POKE J,0:NEXT at the beginning of every 64 program that uses sound. You can also execute the statement in direct mode by typing it in without a line number and pressing RETURN. Incidentally, since the Commodore 128 emulates a 64 when running in 64 mode, it suffers from the same problem; however, in 128 mode RUN/STOP-RESTORE seems to clear the SID chip correctly.

## Hex Keypad For Apple MLX

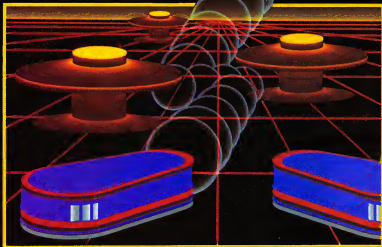
Like Larry Watkins ("Readers' Feedback," May 1985), I usually enter MLX machine language listings with one hand and follow the code with the other. Is it possible to write a program for the Apple IIc that changes the keys to a numeric keypad? I'd like to see a hexadecimal arrangement and a colon you don't have to shift.

Bill Pearson

Only two line changes are required to redefine part of the keyboard as a 16-key hexadecimal keypad for "Apple MLX" (which first appeared in the June 1985 issue and is published periodically in COMPUTE). Replace line 410 of MLX and

# TAKE A BREAK!

For  
Apple, Atari,  
& Commodore 64



## WITH NIGHT MISSION **PINBALL**

You deserve the best. You've earned it. Now reward yourself with a session of Night Mission PINBALL, the most realistic and challenging arcade simulation ever conceived! ■ Stunning graphics and dazzling sound effects put Night Mission PINBALL in a class by itself. Game features: multi-ball and multi-player capabilities, ten different professionally designed levels of play, and an editor that lets you create your own custom modes. ■ So take a break with Night Mission PINBALL from SubLOGIC. Winner of *Electronic Games* magazine's 1983 Arcade Award for Best Computer Audio/Visual Effects.



**See your dealer . . .**

or write or call for more information

**Order Line: 800 / 637-4983**

**subLOGIC**

Corporation  
713 Edgebrook Drive  
Champaign IL 61820  
(217) 359-8482 Telex: 206995

add line 415 as shown here;

```
410 FOR I=1 TO 17: IF K <> ASC(
MID$( "M,/JKLUIOP7890",
I, 1)) THEN NEXT: GOTO 420
415 AS = 15 + MID$( "01234567
90ABCDEF", I, 1) + RS: P = P + 1
```

Once these changes are made, Apple MLX accepts 7-8-9-0 for C-D-E-F, U-I-O-P for 8-9-A-B, J-K-L-; for 4-5-6-7 and M-,-,-/ for 0-1-2-3. You can even put stick-on numbers on the front of the redefined keys. Since you don't need to type colors in Apple MLX listings, the color key has been left alone.

## Better Atari Color Combinations

Please tell me the proper POKE or SET-COLOR command to make my Atari 800XL's text blue on darkest blue, blue on black, white on black, or any other combinations that might be easier on one's eyes. Will leaving the computer in these modes for long periods of time damage anything? Is there any way to make DOS 3.0 work in these altered text colors?

Job Branham

It's quite easy to change the Atari screen colors. The statement SETCOLOR 1, color, brightness sets the brightness level of text. The color value is irrelevant, since text is always the same color as the background (simply a different shade). The brightness value must be an even number from 0 (darkest) to 14 (brightest). Use SETCOLOR 2, color, brightness to control the background color. The color value can be any number from 0-15, and the brightness can be any even number from 0-14. You can also set the border color with SETCOLOR 4, color, brightness.

For instance, the statements SETCOLOR 1,0,10: SETCOLOR 2,0,0 produce light blue on dark blue. Blue text on a black background is not possible, since both screen and character color must be the same color. However, some shades of the same color look like different colors. For example, bright red-orange looks like yellow, and white is actually "bright black." Thus, SETCOLOR 1,0,10: SETCOLOR 2,0,0 gives you white text on a black screen.

Many people find it easier to read black text on a white background, since this combination simulates the appearance of type on paper. Use SETCOLOR 1,0,2: SETCOLOR 2,0,10. You may have to fiddle with the brightness numbers to get the contrast right. Unfortunately, these color changes are transient. The normal screen colors return when you press SYSTEM RESET, change graphics modes, or go to DOS. To change the screen colors of the DOS menu, you'd have to disassemble DOS to find the instruction which sets the colors and then alter the

instruction yourself.

No color combination will damage your TV or monitor unless you leave very bright text on the screen for a significant period of time (such as overnight). Atari computers have a built-in protection feature against burn-in: If you don't press any keys for about nine minutes, the computer automatically enters attract mode, in which the screen colors continually cycle at 50 percent brightness until you press a key.

## Programming The VIC/64 User Port

I have built a breadboard system and interface to the VIC-20 user port, but am having trouble with programs to make use of it. Could you give me more information on how to program the user port?

John W. Farrow, Sr.

The user port, located on the back of the computer on the left side, gives you direct access to the computer and allows control of external parallel and RS-232 serial devices. Access to the user port is through the VIA (Versatile Interface Adapter) chips on the VIC, and the CIA (Complex Interface Adapter) chips on the 64.

Communications with RS-232 serial devices like modems are provided for in the computers' operating system via device 2, so we assume your homebrew interface makes use of the user port's eight-bit parallel data port. The parallel port can be controlled directly from BASIC with PEEK and POKE commands. When the port is being used for input, the address (37136 for the VIC, 56577 for the 64) is PEEKed. When the port is used for output, the address is POKEd.

Before data can be exchanged through the port, the function of the eight data lines must be specified by setting the data direction register for the user port (37136 for the VIC, 56577 for the 64). Each of the eight bits at this address controls the direction of data flow for the corresponding bit of the user port. When a bit in this register is set to 0, the corresponding bit in the user port is used for input. Setting a bit in the data direction register to 1 indicates that the user port bit will be used for output. Pressing R/N/STOP-RESTORE or powering up initializes all bits in the direction register to 0, setting all lines of the port for input. POKEing a value of 255 into the register will set all lines for output. Any combination of input and output lines can be specified by POKEing the value for the desired pattern of 1's and 0's into the data direction register.

Once the data direction register is set up, the desired lines of the user port can be read from or written to by PEEKing or POKEing the data register. If a line is

selected for data input, the corresponding bit in the data register will hold a 0 if the line is at its low state (0 volts) and a 1 if the line is at its high state (at least 2.4—but not more than 5—volts). If the port is set for output, setting a bit in the data register to 0 causes the corresponding line on the port to be set to its low state, 0 volts. Setting a bit to 1 causes the voltage on the line to rise to its high state (usually about +5 volts). For example, the following statements set all eight lines of the VIC's user port for output, then present a high (+5V) state on each line:

```
10 POKE 37138,255
20 POKE 37136,255
```

Since applying improper voltages to the lines of the user port can damage the VIA and CIA chips—rendering your computer useless—we recommend that you use caution when experimenting with the port. If you're unfamiliar with the basics of electronics, you should connect only circuits designed by knowledgeable technicians.

For more information, and a simple peripheral device which can be controlled by the user port, refer to Chapter 5 in *COMPUTE!'s First Book of Commodore 64*. Additional information can also be obtained from Mapping the VIC, Mapping the Commodore 64, and Programming the VIC, from *COMPUTE! Books*.

C

## COMPUTE! Subscriber Services

Please help us serve you better! If you need to contact us for any of the reasons listed below, write to us at:

COMPUTE! Magazine  
P.O. Box 914  
Farmingdale, NY 11737

or call the Toll Free number listed below

**Change Of Address.** Please allow us 6-8 weeks to effect the change, send your current mailing label along with your new address.

**Renewal.** Should you wish to renew your **COMPUTE!** subscription before we remind you to, send your current mailing label with payment or charge number or call the Toll Free number listed below.

**New Subscription.** A one year (\$24.00) US subscription to **COMPUTE!** is \$24.00 (2 years, \$45.00, 3 years, \$65.00. For subscription rates outside the US, see last page). Send us your name and address or call the Toll Free number listed below.

**Delivery Problems.** If you receive duplicate issues of **COMPUTE!**, if you experience late delivery or if you have problems with your subscription, please call the Toll Free number listed below.

**COMPUTE!**  
800-334-0868  
In NC 919-275-9899

# CLOSE ENCOUNTERS OF THE FANTASY KIND



**QUESTRON™** A role-playing game so spell-binding, it suspends reality and conjures up an enchanted world whose only hope against the forces of evil is you.

**GEMSTONE WARRIOR™** A lightning-fast action strategy game that will leave you breathless with excitement as it takes you through the labyrinths of the Netherworld on a noble quest.

Open them up and step through the gateway to your fantasies come true.  
At your local computer/software or game store today.

**QUESTRON™** is available on 48K diskette for APPLE® II with AppleSoft ROM, Apple II+, IIc, and IIx (\$49.95).  
On 40K diskette for ATARI® home computers (\$49.95).  
On 64K diskette for COMMODORE 64™ (\$39.95).



**GEMSTONE WARRIOR™** is available on 48K diskette for APPLE® II with AppleSoft ROM, Apple II+, IIc, and IIx (\$54.95).  
Also on 64K diskette for COMMODORE 64™ (\$54.95).

© 1985 by **STRATEGIC SIMULATIONS, INC.** All rights reserved.

If there are no convenient stores near you, VISA & M/C holders can order direct by calling 800-227-1617, ext. 335 (toll free). In California, 800-772-3545, ext. 335. Please specify computer format and add \$2.00 for shipping and handling.

To order by mail, send your check to: STRATEGIC SIMULATIONS, INC., 660 Siskiyoun Road, Box A-200, Mountain View, CA 94043. (California residents, add 7% sales tax.) All our games carry a "14-day satisfaction or your money back" guarantee.

**WRITE FOR A FREE COLOR CATALOG OF ALL OUR GAMES.**

APPLE, ATARI and COMMODORE 64 are trademarks of Apple Computer, Inc., Atari, Inc., and Commodore Electronics, Ltd., respectively.

# Report From The Summer Consumer Electronics Show

Tom R. Halfhill, Editor

*Considerably less frantic than past Consumer Electronic Shows—at least in the computer section—this summer's CES nevertheless showcased some groundbreaking new products. Foremost was Atari's announcement of a mass storage device that may bring optical memory into homes, schools, and businesses by early 1986.*

**I**t would border on the outrageous to describe any Consumer Electronics Show as "quiet"—considering that 80,000 to 100,000 retailers, wholesalers, middlemen, and journalists spend four days jamming their way into convention halls for what is billed as the world's largest industry trade show.

Still, something was noticeably different about this June's CES in Chicago. The annual noise which emanates from the personal computer section in the McCormick West building had dissipated to a muffled roar.

Only two U.S. computer manufacturers were in attendance: Commodore and Atari. IBM and Apple, as is their custom, skipped the show. Atari, which a few weeks earlier had announced it was pulling out of CES, was enticed back by the show management but occu-

pled a couple of meeting rooms instead of its extravagant exhibit of days past. And the lower level of McCormick West, once the exclusive domain of a hundred computer software companies, now was half-filled with videotape exhibits and purveyors of video porn. Rarely have the effects of the much-publicized industry shakeout been so apparent.

On the bright side, the mood was just slightly more optimistic as both Atari and Commodore moved closer to shipping actual production models of their latest personal computers. In fact, as the show opened, Atari said it had delivered the first 5,000 of its new 520STs to Canada and Europe and was expecting large-volume shipments to the U.S. by July 8. Commodore said it was only weeks away from shipping the Commodore 128, and was gearing up for a press conference in late July to officially announce its eagerly anticipated Amiga computer (see the accompanying article, "A tantalizing Peek At The Amiga").

Several software companies announced new products for the Commodore 128, Amiga, and Atari ST series, although most seem to be cautiously hanging back until they see how the machines fare.

On the hardware front, Atari announced a mass-market version of the ST. Commodore exhibited a dual disk drive for the Commodore 128, and a British company announced it would export a 128K computer to the U.S. But the most interesting news from the show was a revolutionary new peripheral displayed for the first time by Atari:

A very fast mass storage device that uses high-density optical discs. It was a dramatic demonstration that the upcoming generation of personal computers will place much of the power of a mainframe computer on a desktop. It also showed that the personal computer industry is not only far from finished, but is just getting started.

**A**tari's big announcement was the CD-ROM, which stands for Compact Disc-Read Only Memory. If you've never heard of a CD-ROM before, prepare to read dozens of articles about it in coming months, because within two years CD-ROMs will probably be everywhere. (See the accompanying article, "Monster Memory.")

Atari hooked up a working prototype of a CD-ROM to a 520ST on the second morning of the show, and the Atari exhibit was packed for the next three days. Although the hardware looked rough, the software appeared to be nearly complete. In fact, Atari hopes to have the product ready for sale by late this year or early 1986 for about \$500.

The software that allows the CD-ROM to work with the 520ST is being developed by an optical type-setting company, Activenture, Inc. of Pacific Grove, California. Activenture has placed an entire 23-volume, nine-million-word encyclopedia plus index on a single optical disc—and the disc is still three-quarters empty. The system is so efficient that any entry can be looked up in a matter of seconds.

When the CD-ROM is introduced, Atari and Activenture hope to have other databases available, too. Some examples might be additional encyclopedias, legal and medical references, cookbooks, phone books, and the whole Library of Congress card catalog. In



fact, for reference works, the CD-ROM may be a significant step toward the paperless information age.

"The ink-on-paper business was fairly restrictive," says Tom Rolander, vice president for engineering of Activature. "The only way you could look at information was in the way the original publisher had organized and presented it while laying it down on paper. When using reference material—which is why we're starting with encyclopedias and things like that—the degree of usefulness is based on how well we can find the information within that material. In other words, how good its indexing system is. What we have here, by connecting the computer with the CD-ROM, is the ultimate indexing tool. We'll know every reference there is to everything that's in the database."

The paperless information age, however, is encountering some resistance by those whose business depends on putting information on paper—traditional publishers. For example, Atari and Activature had to delay announcing the name of the encyclopedia on their CD-ROM disc because the publisher didn't want any publicity until the final contract was signed. Rolander says all the publishers will probably fall in line once the first one does, but that many publishers are wary of the new technology.

"They can charge \$1,000 for a shelf-full of books, but how much will people be willing to pay for the same information on one of these?" asks Rolander, spinning a disc on his finger. "To consumers, it doesn't look like they're getting as much. Will people pay \$500? Or \$150? Or \$50? We don't know yet. This may finally determine the true value of information."

**B**esides the CD-ROM, Atari announced two new variations of its 520ST, the 260ST and 260STD. They're identical to the 520ST except for three features: 256K RAM instead of 512K; a built-in RF modulator so they can be plugged into an ordinary TV set as well as color or monochrome monitors; and the operating system and GEM (Graphics

Environment Manager) in ROM instead of in RAM. Also, the 260STD has a built-in 3½-inch floppy disk drive. They are scheduled for release this fall for \$399 and \$499, respectively.

Atari says it plans to sell the 260ST and 260STD through mass-market channels, such as discount stores, while restricting the 520ST to specialty dealers, such as computer shops. This is a reversal of Atari's original plan to sell the 520ST through all types of outlets.

Atari also has been wavering back and forth on whether the operating system and GEM will ever be in ROM on the 520ST. When the 520ST was first announced in January, Atari said all the system software would be in ROM. But delays in debugging the operating system prompted Atari to release it on disk with initial shipments of the computer. Then there were conflicting statements about whether early 520ST owners would be able to upgrade to a ROM-based operating system later. (A RAM-based operating system takes longer to boot up, but is easier to revise; a ROM-based system boots up instantly, but can be upgraded only by replacing the ROM chips.)

At CES, Atari President Sam Tramiel told COMPUTE that the operating system will be transferred to ROM for the 260ST/260STD, and then made available at "minimal cost" for 520ST owners with RAM-based systems. "These 18 guys back in Sunnyvale [at Atari's software department] are right now crunching the code to get it into the ROM size [192K]," said Tramiel. "TOS [Tramiel Operating System] now I think is 205K, or something like that. We feel it's not a big problem, but we've got to get it done fast."

**C**ommodore was relatively idle at this CES. Its only new hardware announcements for the U.S. market were a dual floppy disk drive for the Commodore 128 and a dot matrix printer. Interestingly, before CES started, Commodore intended to show a hard disk drive for the 64 and 128, but pulled the product at the last minute for unknown reasons. Commodore also pulled the

LCD portable lap computer first shown at the January CES; reportedly, the machine has been postponed while Commodore concentrates on bringing the 128 and Amiga to market.

The new 1572 dual disk drive combines two 1571 drives in a slim-line case designed to sit atop the Commodore 128. It has the same multimode capabilities as the 1571 (Commodore 64, Commodore 128, and CP/M formats). Commodore says it should be available this summer, but no price was announced.

The new MPS 1000 is a multimode dot matrix printer. In draft mode, it prints at 100 characters per second (cps); in near-letter quality mode, it prints sharper characters at 16 cps; and in graphics mode, it has a density of 50 to 240 dots per inch. It can also print in widths ranging from 80 to 160 columns. It's compatible with the Commodore 128, 64, and many other personal computers. Like the 1572 disk drive, it's scheduled to be available this summer, but no price was announced.

Two products exhibited at the Winter CES were firmed up at this show. The Commodore 1670 direct-connect modem, which transfers data at 1200 bits per second, will sell for around \$200 and has auto dial, auto answer, auto mode selection, and auto speed switching from 300 to 1200 bps. It works with the 128, 64, SX-64, Plus/4, and VIC-20. And the Commodore two-button mouse controller first seen in January will sell for \$49.95 and should be available immediately. It works with the 128, 64, and VIC.

Commodore also showed four interesting computers for foreign markets, but apparently they won't be available in the U.S. in the near future. The Commodore 128D Integral Personal Computer is a variation on the 128 that separates the keyboard from the system unit and includes a built-in disk drive. Commodore says it will be available in Europe late this year. The PC10 and PC20 are IBM-compatible computers recently introduced in Europe. The PC10 has 256K RAM and two 360K floppy disk drives; the PC20 has 512K RAM, one floppy drive, and a ten-megabyte hard disk. And finally, the Commodore 900 Business Computer is a multitasking, multiuser workstation that uses a

# Monster Memory

**T**he CD-ROM, an acronym for *Compact Disc-Read Only Memory*, is a compact disc audio player which has been slightly modified for general-purpose data storage and interfaced to a computer.

Compact disc players are the latest rage among audiophiles. Up to 75 minutes of digitally encoded music can be stored in the form of microscopic pits on a 4.7-inch rigid plastic disc. Inserted in a special player, the disc spins at 300 r.p.m. while a miniature laser reads the pits. The data is decoded by a microprocessor, then converted into standard audio signals which are fed into the auxiliary input or tape monitor jacks on a stereo receiver. The result is exceptionally pure music of unprecedented dynamic range and frequency response, free of surface noise and tape hiss. Furthermore, since the disc is read by a laser, not a diamond stylus, compact discs last virtually forever with no deterioration. They can also tolerate rougher handling than ordinary records and tapes.

But music isn't the only thing a compact disc can store. Any type of information can be digitized and recorded on a disc. That includes text, graphics, and computer programs. And the capacity is enormous: A single compact disc stores about 550 megabytes. A megabyte equals 1,024K, so that's roughly equivalent to 1,564 floppy disks on an IBM PC, 3,520 disks on a Commodore 1541 drive, 4,022 disks on an Apple II, 4,469 enhanced-density disks on an Atari, or 6,400 single-density Atari disks. They're cheap, too: compact discs can be mass-produced at a manufacturing cost of a few dollars each (audio discs currently retail for about \$15). Because audio CDs and CD players are already in mass production, CD-ROMs can debut at affordable prices.

A compact disc is a read-only storage medium, so you can't record data on it yourself. But CDs are ideal for storing large databases that don't have to be updated often. At CES, Atari demonstrated a sample disc that contained a 23-volume, nine-million-word encyclopedia. The encyclopedia was transferred to the CD from magnetic tape, where it was stored in punchcard format—the equivalent of 976,000 punchcards. Yet, it fits on one-quarter of the space of a single CD.

To think of a CD simply as an efficient way to store mass amounts of information is to miss the point, however. Like a floppy disk drive, a CD player is a *random-access* device; it can seek and retrieve any piece of data on the disc in a few seconds without hunting through the information sequentially. Therefore, a CD-ROM can find the slightest, most obscure fact in a massive database in less time than it takes you to pull a book off a shelf and flip it open to the index.

**H**ere's an example: Let's say you're a student researching a report on Thomas Jefferson. On the Atari CD-ROM, there are two ways to approach the task.

The first way is very similar to the usual method of looking up something in an encyclopedia. First, you boot up the CD-ROM on the Atari 520ST. This takes only a few seconds. A graphics display on the screen shows a bookshelf with a 23-volume encyclopedia. By rolling the ST's mouse controller, you move the screen pointer to the "J" volume and then click the mouse button. This brings up another screen with a graphics picture of the book you selected, opened to several alphabetical tab entries. You move the pointer to the tab which would include *Jefferson*—for in-

stance, *Japan to Jet*. Another click calls up a screen showing all the article titles within that section. When you move the pointer to the title *Jefferson, Thomas* and click the button again, the computer loads the article (and several preceding and following articles, as well) from the CD into memory. It takes less than four seconds for the CD-ROM to fill the 520ST's entire 512K RAM.

Now you can read the article on the screen, scrolling or flipping pages by clicking the mouse. This method of looking up subjects is recommended for those who aren't familiar with computers, because it requires almost no computer knowledge.

The second method takes greater advantage of the computer's power. Instead of looking up the subject alphabetically by yanking a graphics image of a book off a shelf, you pull down a menu and select the *search screen*. This screen presents a number of options; to keep it simple, you can just type *Thomas Jefferson* at the prompt and ask for a general search. In about four seconds, the computer reports how many times the phrase *Thomas Jefferson* appears in the encyclopedia. You can flip to the first occurrence by clicking the mouse. Again, in less than four seconds, the computer loads the article from the CD into memory and even highlights your search phrase within the text. You can flip to subsequent occurrences merely by clicking the mouse button.

What makes this technique so powerful is that the computer will find references to Thomas Jefferson in articles that may never have been checked using the old-fashioned method. The student may learn that Jefferson was not just a politician, but also an inventor, architect, and connoisseur of wines. Looking up the same references in even the most thoroughly cross-indexed paper encyclopedia would be much more time-consuming.

When the Atari CD-ROM software is finished, it will allow two-dimensional searches, too. You could look up every article that contains references to Thomas Jefferson and Thomas Paine, or Thomas



Get the jump on the weatherman by accurately forecasting the local weather yourself!



A scientifically proven way to develop an awesome memory.



You are trapped in a five-story, 120-room structure made entirely of ice. Find the exit before you freeze!



Take control of your personal finances in less than one hour a month.



The beautiful princess is held captive by deadly dragons. Only a knight in shining armor can save her now!



Cut your energy costs by monitoring your phone, electric and gas bills.



Computerize car maintenance to improve auto performance, economy and resale value.



Create multi-colored bar graphs with a surprisingly small amount of memory.



A time-saving organizer for coupons, receipts and more.



School-age and pre-school children are rewarded for right answers, corrected on their wrong ones.



A real brainflexer. Deflect random balls into targets on a constantly changing playfield.



A fun way to dramatically increase typing speed and accuracy.

## Get up to 30 new programs and games for less than 15 cents each—every month in COMPUTE!

Every month, COMPUTE! readers enjoy up to 30 brand-new, ready-to-run computer programs, even arcade-quality games.

And when you subscribe to COMPUTE!, you'll get them all for less than 15 cents each!

You'll find programs to help you conserve time, energy and money. Programs like Cash Flow Manager, Retirement Planner, Coupon Filer, Dynamic Bookkeeping.

You'll enjoy games like Air Defense, Boggler, Slalom, and High Speed Maze.

Your children will find learning fast and fun with First Math, Guess That Animal, and Mystery Spell.

Looking for a challenge? You can write your own games. Customize BASIC programs. Even make beautiful computer music and pictures.

It's all in COMPUTE!. All ready to type in and run on your Atari, Apple, Commodore, PET/CBM, TI-99/4A, Radio Shack Color Computer, IBM PC or IBM PCjr.

What's more, you get information-packed articles, product reviews, ideas and advice that add power and excitement to all your home computing.

And when it's time to shop for peripherals or hardware, check COMPUTE! first. Our product evaluations can save you money and costly mistakes. We'll even help you decide what to buy: Dot-matrix or daisy-wheel printer? Tape storage or disk drive? What about modems? Memory expansion kits? What's new in joysticks, paddles and track balls?

Order now! Mail the postpaid card attached to this ad and start receiving every issue of COMPUTE!.

**For Faster Service  
Call Toll-Free  
1-800-334-0868**

Jefferson or George Washington. Other options let you limit the search for occurrences in adjacent words, single paragraphs, and word groupings of various sizes.

**I**f you're an experienced programmer, you might be puzzled by the search times described above. Even in superfast machine language on the 520ST's 68000 microprocessor, how can the computer search through nine million words in less than four seconds?

The answer is that the computer can't. Instead, it refers to an extremely sophisticated index of search tables on the CD.

The search tables were compiled with a VAX minicomputer by Activision, Inc., the company which is developing the CD-ROM software for Atari. First, the VAX built a dictionary by identifying every unique word in the encyclopedia—more than 140,000 words. Then it compiled search tables which cross-reference the dictionary to every occurrence of each word in the encyclopedia. In conventional book form, the finished

index would occupy about 20 volumes. That means the index is nearly as long as the encyclopedia itself.

When you request a search, the 520ST simply consults the index of search tables on the disc and rapidly locates each occurrence of the search phrase. In effect, the searching has already been done for it by the VAX.

Retrieving the information is very fast, too, though not quite as fast as with hard disks. Because CDs were designed for storing music, which is played sequentially, their random-access capabilities are not as good as they could be. Still, they're much faster than most floppy disk drives. Data is stored on a CD in 270,000 records containing 2,048 bytes each; the average access time for a record is about one second. The greatest access time—which happens when the head must move from an extreme outside track to an inside track or vice versa—is three to four seconds. To keep this from happening very often, data is recorded on a CD on the inside tracks first, moving outward. This takes advantage of the CD-ROM's very rapid track-to-track access time. To read a nearby

track, the player merely tilts a tiny mirror to refocus the laser rather than repositioning the entire head. (Each track of microscopic pits is only two microns—two millionths of a meter—wide).

Once the information is located, the CD-ROM feeds it to the computer at a rate of 75 records (150K) per second. To make sure the data arrives without errors, the CD-ROM's error-correction capabilities have been improved over that of a regular audio CD player. An audio player can be expected to pass one bad bit for every 10,000 bits—inadvisable when listening to Beethoven or the Beatles, but not nearly good enough for computer storage. So CD-ROMs employ an error-checking and correction scheme that allocates an additional 288 bytes for each 2,048-byte record. That much overhead—more than 14 percent—would be wasteful on a floppy or hard disk, but CDs have room to burn. The improved error rate on a CD-ROM is at least as good as with a hard disk: about one bit error for every 1,000,000,000,000 to 1,000,000,000,000,000 bits (one trillion to one quadrillion).

#### Report From Summer CES Continues . . .

Unix-compatible operating system.

Although some of these higher-end computers will be available in Canada, Commodore announced no plans at present to market them in the U.S.

**A**nother new personal computer was announced at CES by a British company, Amstrad. Already available in Europe, where several hundred thousand units have been sold, the Amstrad CPC6128 is scheduled to be shipped to the U.S. later this year.

The U.S. version of the Amstrad has an 8-bit Z80A microprocessor, 128K RAM, a built-in 3-inch disk drive, CP/M compatibility, BASIC and Logo, an expansion interface, joystick port, and stereo sound output. It comes packaged in two configurations. One includes a green-screen monitor and WordStar word processor for \$699, and the other has an RGB color monitor, Amstrad's own word processor, and some entertainment

software for \$799.

Briefly, here are some other highlights of the Summer CES:

- Abacus Software of Grand Rapids, Michigan announced *Super C*, a C compiler for the Commodore 64 and 128. It has a full-screen editor with horizontal and vertical scrolling and is compatible with most other versions of C. Source files up to 41K long can be created.

- Commodore is releasing several titles for the 128, including *June 2.0*, an icon-based integrated package with a word processor, spreadsheet, and filing manager, all of which can be manipulated with the mouse controller; *Micro Illustrator*, a graphics drawing program formerly available for the 64, which takes advantage of the 128's extra memory and other features; and the *Perfect* series for the CP/M operating system, consisting of *Perfect Writer*, *Perfect Calc*, and *Perfect Filer*. All three work in the 80-column mode, are capable of sharing files, and have pop-up menus, split

screens, and automatic formatting for printouts.

- Epyx, Inc. of Sunnyvale, California is releasing *Winter Games*, a sequel to *Summer Games*, for the 64, Apple, and Macintosh; two new LucasFilm games for the Atari, *The Eidolon* and *Koranic Rift*; *The World's Greatest Football Game* for the 64, Apple, and Atari; and the *Temple of Apsai Trilogy* for the 64, Apple, Atari, Macintosh, and IBM. (Prices will range from \$19 to \$35.)

- Batteries Included of Richmond Hill, Ontario is releasing its *PaperClip* word processor for the Apple (\$89.95) and Commodore 128 (\$119.95 with *SpellPak*); the BI-80, an 80-column video adapter on a cartridge for the Atari XL and XE series (\$79.95); an 80-column version of Atari *PaperClip* for the BI-80 (\$59.95); new versions of *HomePak*—a combination word processor, filer, and terminal program—for the IBM PC/PCjr, Commodore 128, Apple, and Atari ST (\$49.95); an IBM version of *The Consultant*

For  
APPLE®  
and  
COMMODORE 64™

When sorcery ruled, and trolls and minotaurs still walked this Earth,  
a party of six intrepid adventurers set out to find the Nine Rings  
and use them to destroy the Dark Lord. This is your quest.  
This is your...

# PHANTASIE™

A MULTIPLE-CHARACTER ROLE-PLAYING ODYSSEY

Look for this new, exciting game  
at your local computer/software  
or game store today!

If there are no convenient stores  
near you, VISA & Mastercard holders  
can order direct by calling 800-227-1877,  
ext. 335 (toll free). In California, call  
800-773-3545, ext. 335.

To order by mail, send your check to:  
STRATEGIC SIMULATIONS, INC., 863 Sterling  
Road, Bldg. A-200, Mountain View, CA  
94043. Please add \$2.00 for shipping  
and handling (California residents,  
add 7% sales tax). All our  
games carry a "14-day  
satisfaction or your  
money back"  
guarantee.

WRITE FOR A  
FREE COLOR  
CATALOG OF  
ALL OUR GAMES.

PHANTASIE is \$39.95.

\$19.95 (discount for Apple II with AppleLink ROM, II+ & IIcx and PC).  
Also available for Commodore 64.\*

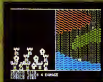
\*Check for compatibility with your version of Apple Computer, Inc.  
ROM. © 1988 Strategic Simulations, Inc. All rights reserved.  
In California, please add 7% for all goods shipped.

# SSI

STRATEGIC SIMULATIONS, INC.



1. Search for the Nine Rings in a lush, scenic landscape.



2. Use your battle strategy skills to defeat the evil forces of the Dark Lord.



3. Use your battle strategy skills to defeat the evil forces of the Dark Lord.



4. Explore the world of Phantasie, a lush, scenic landscape.

# A Tantalizing Peek At The Amiga

**A**t this writing (early June), Commodore is scheduled to unveil its long-awaited Amiga Lorraine computer at a press conference in New York in late July. Although the machine was substantially ready in June, Commodore evidently kept it off the show floor at CES to avoid stealing attention away from the Commodore 128, which was due to begin appearing on store shelves within weeks. Nevertheless, despite unprecedented secrecy and security, more and more details about the Amiga leaked out at CES press parties. Also, COMPUTE! arranged a peek at the Amiga through a helpful source. We learned that even more capabilities have been added since our first look at the computer a year ago at the June 1984 CES (see "Software Power! The Summer Consumer Electronics Show," August 1984).

The Amiga's final configuration was still subject to change before its official introduction, but here's what it will probably include:

- Motorola 68000 microprocessor for the central processing unit. This is the same 16/32-bit chip found in the Apple Macintosh and Atari ST series. It can address up to 16 megabytes of memory (1 megabyte = 1024K).

- 256K of RAM (Random Access Memory), expandable to at least 512K on the system unit and more externally.

- 192K of ROM (Read Only Memory) containing *Intuition*, a Macintosh-like operating system

with pull-down menus, icons, hierarchical disk directories, multitasking, and mouse support. Unlike the Macintosh's operating system, however, *Intuition* can be manipulated with keyboard controls as well as the mouse. The keyboard, which is attached to the Amiga system unit with a coiled cord, includes cursor keys and a numeric keypad.

- True multitasking. Thanks to the high-speed 68000 and a number of dedicated chips for input/output and other vital functions, the Amiga can run several programs simultaneously with no apparent sluggishness, even while simultaneously accessing the disk drive. By opening screen windows of various sizes, you can watch all the programs running at once. This is an especially useful feature for business applications—you could simultaneously work with a word processor, terminal program, database manager, and spreadsheet without stopping one program to start another.

- A built-in double-sided 3½-inch disk drive that stores about 800K per disk. These are the same hard-shell microfloppy disks used by the Macintosh and Atari ST series. Up to three external drives can be added to the Amiga by daisy-chaining.

- Custom chips for graphics and animation. Maximum screen resolution is 640 × 400 pixels (screen dots), with additional graphics modes of 640 × 200 and 320 × 200. Eighty-column text is standard, but the display is adjustable to narrower widths for greater readability on low-resolution screens.

- Composite video/mixer input that allows you to feed standard video signals into the Amiga, display them on the screen, and then superimpose the Amiga's text and graphics. The external video source could be a video camera, videocassette recorder, videodisc player, TV receiver, or even another computer. The potential of this feature is exciting: games and educational programs with superrealistic backgrounds, titles for home videotapes, and so on. Furthermore, a relatively low-cost peripheral called a *frame grabber* lets the Amiga digitize the incoming video signal so it can be manipulated with graphics utilities, stored on disk, and even dumped to a graphics printer.

- A palette of 4,096 colors, more than any other general-purpose personal computer on the market. Up to 32 of these colors can be displayed at once without special tricks.

- Video outputs for TV sets, composite color and monochrome monitors, and high-resolution RGB color monitors. Reportedly, the RGB output is analog, like the Atari ST's, so it's not compatible with RGB monitors designed for the Commodore 128 and IBM PC computers. Analog RGB allows more color intensity levels than IBM-type RGB.

- An expansion port that includes every line on the system bus for almost unlimited expansion capabilities. This could include coprocessors, such as an 8088/8086 board for IBM compatibility. Reportedly, the Amiga will be an "open system." That is, to help independent hardware and software designers access the full power of the computer, Commodore is said to be preparing extremely detailed documentation on all aspects of the Amiga for general release. One insider who has seen the preliminary documentation says it's so complete you could almost build the system from scratch using the information it contains.

- Seven-level direct memory access (DMA) controller. Along with the Amiga's many dedicated chips, this lets the machine perform several tasks simultaneously with

# ONLINE IN THE U.S.A.



Be anything you want on American PEOPLE/LINK, where people from across the country get together every night.

They are people with interests in sports, movies, music, games, and just meeting other people . . . people like you. Whatever your interests or fantasies, you'll find someone to share them with on PEOPLE/LINK.

See for yourself why so many men and women are getting together and partying online with us. Not screen after screen of boring data . . . just lots of fun and friendly people.

And our low, low rates won't turn your online dreams into nightmarish monthly charges. To

discover our affordable videotex entertainment, call or write us today.

PEOPLE/LINK . . . the network that says "Thank You."



American PEOPLE/LINK  
Arlington Ridge Office Center  
3215 N. Frontage Road - Suite 1505  
Arlington Heights, IL 60004

1-800-524-0100 (Toll Free) 1-312-870-5200 (Illinois)

no apparent slowdown. It also simplifies things for programmers. An independent software developer who attended a recent Amiga symposium sponsored by Commodore in Monterey, California, told COMPUTE! that systems-level programming on the computer is so easy that "it's more like parameter-passing than programming. You just decide what you want to do—pull data off the disk, whatever—and pass the appropriate parameters to the operating system, and the OS passes them along to the DMA controller, and everything happens during the 68000's off-phase cycle so the machine doesn't slow down at all." (The programmer's symposium, incidentally, was open only to independent developers and was sealed off by armed guards. Those who attended had to sign a strict nondisclosure agreement.)

- Built-in printer drivers to dump the Amiga's bitmapped screens onto graphics printers, including color and inkjet printers.

- Four-voice sound chip with stereo output. The Amiga's custom sound chip routes two voices to each stereo channel for high-fidelity reproduction through external stereo systems. In addition, the sound chip is the most advanced in any personal computer, surpassing even the Commodore 64's SID chip. The Amiga can closely simulate a wide variety of musical instruments, and at least a dozen instrument sounds are built in—such as guitar, pipe organ, cymbals, drums, piano, and violin. Sound envelopes (attack-decay-sustain-release) can be modified simply by pulling down a menu and making selections with the pointer.

- Digital sound sampling. At

this point, it's not clear whether this feature will be standard or optional. Even if it's optional, however, insiders say it will cost much less than anything similar now on the market (the least expensive high fidelity samplers now cost around \$2,000). Digital sound sampling lets you feed sound from an external source into a computer, convert it to digital format, and then play it back, modify the waveforms, or store it on disk. With this capability, programmers and musicians no longer have to spend hours trying to simulate a musical instrument or sound effect—they can just feed the sound directly into the computer from a record player, tape deck, microphone, or instrument, and then manipulate it at will. In fact, some Amiga software developers are taking this approach to cut down development time.

- Built-in speech synthesis. The operating system includes a speech program with text-to-speech conversion. Since this is a standard feature that requires no extra hardware, you can expect many programs to take advantage of the Amiga's speech capabilities—including programs written by home users. COMPUTE! has heard the Amiga talk, and its pronunciation was more understandable than most speech synthesizers now available for personal computers.

- BASIC programming language. Unlike the Macintosh, the Amiga will come with a language for those who want to write their own programs, but it's unclear at this point if BASIC will be built into ROM or loaded from disk into RAM. Pascal and C compilers will be available soon after the computer's release. A 68000 assembler also

will be offered, but it may not be as popular as assemblers on the current generation of personal computers—the compilers are so fast that few programmers are using machine language. One software developer says he has yet to see a single program written for the Amiga in machine language.

- Hard disk option. Although Commodore is said to have no immediate plans for a hard disk drive, an independent company known for its IBM peripherals is rumored to be preparing a hard disk with 10 to 20 megabytes of storage. This would probably interface to the expansion port.

**I**f you take the word of those who have worked with the Amiga, it is the most powerful personal computer ever assembled. In terms of performance, they say it's more than a match for any business-oriented machine now on the market, and has the advanced graphics and sound features demanded by home users as well. But how much will it cost?


At the Monterey symposium, Commodore reportedly told developers that the Amiga would be sold with an RGB monitor and some software for about \$2,000. Since then, however, Commodore has encountered some negative reaction to that figure. Many observers think a lower price is necessary to dramatically undercut the Macintosh and IBM PC, and also to compete with the Atari ST series for a larger share of the intermediate-priced market. At CES, rumor had the price ranging anywhere from \$1,300 to \$1,900, possibly without an RGB monitor.

*Report From Summer CES Continues . . .*

database manager (\$99.95); *Battery-Pak*, a seven-function desktop accessory for the Macintosh (\$49.95); and *Literature Challenge: Introduction to Shakespeare*, an educational program for the Apple, Commodore 64, and IBM PC (\$29.95). In addition, Batteries Included announced a new line of integrated programs for the Atari ST, Commodore Amiga, and MS-DOS computers with GEM. Called the IS line, all the programs will have

Macintosh-like graphics in color and a number of powerful features. The first program, *Portfolio*, is for stock management and will be released for the IBM PC and Atari ST this fall (\$249.95 for the IBM). Others in the series will be a word processor with built-in spelling checker, a spreadsheet and graphics package, and a database manager.

- Broderbund Software of San Rafael, California announced three

new programs to work with *The Print Shop*, its popular printer utility. They are *The Print Shop Graphics Library: Disk One*, *The Print Shop Graphics Library: Disk Two*, and *The Print Shop Companion*. Other new programs are *Bank Street Filer* and *Bank Street Mailer*, sequels to the *Bank Street Writer* word processor; and *Fantavision*, a special-effects generator for the Apple that uses animation technology adapted from the movie industry. 



# COMPUTE! Books

## Are More Than Fun & Games

They're also challenging. Informative. Full of valuable applications for your home and business. You'll find that our newest top-quality releases contain everything you need for successful word processing, programming, or investment management.

### SpeedScript: The Word Processor for the Atari

Charles Brannon, **\$9.95, (Disk, \$12.95)** ISBN 0-87455-003-3

SpeedScript, the most popular program ever published by COMPUTE! Publications, is a commercial-quality word processor for the Atari 400, 800, XL, and XE computers with a minimum of 24K RAM. Included in this book are all the listings and documentation necessary to use Atari SpeedScript. In addition, we've included information and documentation about SpeedScript that have never been published before. For the price of the book, the user receives a commercial-quality word processor—an exceptional software bargain. There is a disk available for an additional \$12.95 which includes programs in the book.



### Investment Management with Your Personal Computer

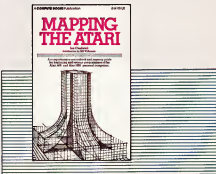
Richard C. Dorf, **\$14.95**, ISBN 0-87455-005-X

Covering the best available investment software for the IBM PC and Apple lines of computers, this valuable guide discusses and evaluates the programs that can help you better manage your investments. This isn't a manual of investment strategies; instead *Investment Management with Your Personal Computer* introduces all the software you'll find vital. Capsule reviews are given for programs ranging from electronic spreadsheet aids and financial databases to portfolio managers and graphics creators. The most significant software is discussed in great detail and thoroughly evaluated.

### Mapping the Atari, Revised

Ian Chadwick, **\$16.95**, ISBN 0-87455-004-1

This classic source book for users of the Atari 400 and 800 has been updated to include the 600/800XL and the new XE series Atari computers. This is the perfect supplement to the owner's manuals, and will prove valuable to both the novice and the expert. As with most COMPUTE! books, *Mapping the Atari, Revised* also contains useful appendices and indices.



For the best in home and office applications, you need COMPUTE! books.

Visit your local computer or book store for best-selling COMPUTE! books, or order directly from COMPUTE!.

Call toll-free 800-334-0868 (in NC 919-275-9809) for more information.

COMPUTE! Books are available in the U.K., Europe, Middle East, and Africa from Holt Saunders, Ltd., 1 St. Anne's Road, Eastbourne, East Sussex BN21 3UN, England.

**COMPUTE! Publications, Inc.** abc  
One of the ABC Publishing Companies



# The Beginners Page

Tom R. Halfhill, Editor

## FOR-NEXT Loops, Part 4

Over the past few columns, we've covered some of the useful techniques possible with the FOR-NEXT statement. Even if FOR-NEXT could do nothing else than what we've demonstrated, it would be indispensable. Fortunately, it can do still more—and all it takes is an optional keyword, STEP.

With STEP, you can construct "long-legged" loops and counters by varying the step size by which the counter variable is incremented. You can even make FOR-NEXT loops that count *backward*. For the sake of illustration, let's say you want to print out all the odd numbers less than 100. Without a FOR-NEXT loop, you could take this approach:

```
10 X=1
20 PRINT X
30 X=X+2
40 IF X<100 THEN 20
```

The alternative is a little shorter and easier to follow:

```
10 FOR X=1 TO 100 STEP 2
20 PRINT X
30 NEXT X
```

Without the STEP option in the FOR statement at line 10, this program would just print all the numbers from 1 to 100 because the counter variable X would be incremented by one during each pass through the loop, as usual. STEP 2 simply tells the computer to increment the counter variable X by two during each pass through the loop. That is, when the loop begins, X equals 1. After the first pass, X equals 3. After the second pass, X equals 5, and so on.

### No Cause For Alarm

Interestingly, although the FOR statement in line 10 tells the computer to count from 1 to 100, the counter variable X actually reaches 101. You can verify this by typing PRINT X and pressing RETURN or

ENTER after the program finishes. The computer reports the final value of X is 101. But don't be alarmed—the computer isn't being disobedient. Although the final STEP 2 increases X from 99 to 101, the computer still performs only 50 loops, since the upper limit specified in the FOR statement is 100 and we're stepping by twos. The program works the same if you change line 10 to FOR X=1 TO 99 STEP 2.

For even more flexibility, the STEP option lets you loop in steps of *any* increment, including fractions and negative numbers. All of the following FOR statements are valid:

```
FOR X=1 TO 1000 STEP 10
FOR X=15 TO 25 STEP 0.5
FOR X=100 TO 1 STEP -1
FOR X=1 TO 0 STEP -0.1
```

It may not be immediately apparent why you'd want to make such strange-looking loops. Mathematical operations are one typical application, but beyond that it's hard to generalize. This kind of loop is generally used to solve certain programming problems. For instance, to make a musical note decay on an Atari, you could gradually reduce the volume parameter of the SOUND statement with a backward loop (FOR X=15 TO 0 STEP -1: SOUND 0,200,10,X: NEXT X). If the note decays too quickly, you could slow it down by reducing the volume by smaller steps (STEP -0.5 or STEP -0.2) rather than embedding a second delay loop.

As you write more programs and use FOR-NEXT loops more often, eventually it will come to you in a flash that a fractional- or backward-stepping loop is exactly the solution to your problem.

### BASIC Variations

FOR-NEXT statements don't follow

the same rules on all computers, so you might need to consult your BASIC manual. In general, Microsoft BASICs (built into Commodore computers, the Apple, IBM, and others) let you omit the variable name from the NEXT statement if you want to. Instead of entering NEXT X you can just type NEXT. This makes the loop run faster. You can also close nested loops in Microsoft BASIC with a statement such as NEXT Z,Y,X instead of NEXT Z:NEXT Y:NEXT X or NEXT: NEXT: NEXT. (These options are not available in TI BASIC or Atari BASIC.)

Try to avoid jumping out of FOR-NEXT loops with GOTO and GOSUB statements. It's considered bad programming form, partly because it makes the program hard to follow. Besides that, a program that repeatedly exits loops with GOTO or GOSUB before the loops are finished can eventually cause some computers to crash with an out-of-memory error or the like. A few versions of BASIC have a special statement that lets you exit a loop with GOTO or GOSUB without causing any problems. In Atari BASIC the statement is POP; it's not available in Microsoft BASIC.

Almost all BASICs require the counter variable in a FOR-NEXT loop to be an ordinary numeric variable; array variables and integer variables are not allowed. An exception is IBM BASIC, which does permit integer variables.

C

The complete 1541 enhancement system ...

# STARDOS

**MAKES  
GREAT COFFEE!!**

The 1541 is the slowest disk drive on planet earth. Even simple operations seem to take forever. Quickloaders and Fastloaders that software-patch the operating system are vulnerable to being knocked out of memory, rendering them totally useless. Even Flashier products that require permanent modifications to the 64 and 1541 can't compete with the blinding speed of STARDOS.

STARDOS accelerates every (yes, we said every) function of the 1541 disk drive. Other fast loaders only load PRG files faster. STARDOS also speeds up SEQ, REL, USR and DIRECT ACCESS files. Everything including FORMAT, VERIFY, SCRATCH, VALIDATE, INITIALIZE and COPY are much faster. In addition STARDOS adds a vast array of easy to use commands and utilities all at the touch of a key.

**A sampling of STARDOS features:**

- Accesses ALL types of files up to 1,000% faster!
- Saves up to 300% faster than normal (with extended verify)
- DOES NOT CHANGE THE SPEED AT WHICH THE DRIVE MOTOR SPINS
- Makes your 1541 MORE RELIABLE and LESS PRONE TO BREAK DOWN or OVERHEAT!
- STARDOS is fully expandable for multiple fast disk drives
- Easy (5 minute) plug in installation. User friendly manual
- 100% compatible with software and serial bus peripherals
- Adds years of life to your disks and drive in reduced wear
- Cures a number of bugs in the Commodore 64 and 1541 disk drive including:
  - The damaging 'Head Knock' that can mis-align your 1541
  - The @: Save with replace bug!!
  - The Editor lock-up bug
- Harness the full power of your disk drive with the built in DOS wedge
- Lock/unlock files and protect/unprotect disks from the keyboard
- Powerful sector editor allows direct viewing of diskette
- Upgrades computer and drive to the latest Commodore specifications
- Instant access to the built in mini-wordprocessor for short notes and memos
- Built in copier copies all file types (even relative) easily and effectively
- Built in disk duplicator copies an entire diskette in less than 3 minutes
- Fully expanded machine language monitor, always on line
- Quality hardware. Full 120 day repair or replace warranty
- Satisfaction guaranteed or money back—no questions asked
- (Oh, by the way, we lied, STARDOS makes LOUSY COFFEE)

**Simply the best. \$74.95**

**[Take advantage of our introduction special at \$64.95]**

Personalize your C64! for a \$10 (non-refundable) charge we will include a 21 character power-on message with your favorite screen, border and text colors. Example:

Background Black Border Red Text Green

*Write or phone for additional information.*

\*Trademark of Commodore Business Machines

WRITE OR PHONE

## STARPOINT SOFTWARE

Star Route 10

Gazelle, CA 96034

[916] 435-2371

When ordering by mail

• \$64.95 + 3.00 shipping

• \$64.95 + 4.00 COD orders

• Shipping out of USA \$6.00

• Calif. residents add 6% sales tax

• VISA or Mastercard accepted

Please allow 4-6 weeks for delivery



## Compilers, Interpreters, And Flow: Part 2

Last month I argued that one of the reasons Logo isn't used for serious applications programming is because it's not generally available as a compiler. I'm not suggesting that programmers shouldn't have interpreters—they should. However, I am suggesting that the ideal programming environment might include an interpreter for writing and testing programs, and a compiler so the completed (and mostly debugged) programs can execute much faster.

There are many fine Logo interpreters on the market, but—as this is written—only one Logo compiler: *ExperLogo* from Expertelligence (for the Apple Macintosh). To illustrate the difference between these two Logo environments, let's examine a program that uses recursion to create a fractal "sponge." (If you're interested in programming fractals, by the way, you might want to explore the subject further in two of my books, *Discovering Apple Logo* and *Beyond Turtle Graphics*, both published by Addison-Wesley.)

Here's how the program is written with an interpreter, *Apple Logo II*, running on a 128K Apple IIe or IIc:

```
to sponge:size:limit
  ifsize <:limit [forward:size stop]
  sponge:size:/3:limit
  left 60
  sponge:size:/3:limit
  left 60
  sponge:size:/3:limit
  right 120
  sponge:size:/3:limit
  right 120
  sponge:size:/3:limit
  left 60
  sponge:size:/3:limit
  left 60
  sponge:size:/3:limit
end
```

Once this procedure has been entered, it can be executed by enter-

ing its name with the appropriate values chosen for the variables. For example, the basic motif for the curve can be seen by entering:

```
right 90
sponge 81 81
```



To see a more detailed level of this curve, we could clear the screen and enter:

```
right 90
sponge 81 3
```



With the Logo interpreter, this picture takes 223 seconds to complete. However, as soon as the program is written it can be executed; there is no time delay before the program starts to run.

## Interactive Rhythms

To write this program in *ExperLogo*, you enter the same source code into an edit window on the Macintosh screen. Then the code is selected and compiled. The compilation time for this program is 19 seconds on a 128K Macintosh (faster on a Fat Mac with 512K).

Once the program is compiled, it can be executed with the commands shown above. The compiled program draws the picture in 7 seconds—more than 30 times faster than the interpreted Logo. Of course, the compiled program executes faster partly because it's running on a 16/32-bit computer

rather than on an 8-bit computer; but, as users of Macintosh BASIC will attest, interpreters can run slowly even on the Macintosh.

Is the speed gain important? For small programs, it may not be. But humans are funny creatures. We have certain rhythms in our interactions with each other and with our machines. If our technology is not operating at our pace, we become frustrated. For example, even though most photocopy machines operate pretty quickly, the perceived difference between a copier that takes 10 seconds for a copy and one that takes only 2 seconds is quite large. Those 8 seconds are just long enough to destroy the sense of flow.

In the case of computer programs we use every day, this sense of flow is even more important. When experimenting with graphic images such as those shown above, the sense of interactivity—of being able to tinker with the curves—is lost when each picture can take several minutes to create.

The Logo compiler from Expertelligence is most welcome, since it allows programmers to write commercially useful software with a powerful language—a language that in its interpreted form is often perceived as just a tool for children to draw pretty pictures.

Next month, I'll show that just as a compiler has made Logo a much more useful language, an interpreter is having the same effect on a language for which compilers are the norm—Pascal.

Thornburg welcomes letters from readers, but regrets that he cannot always provide personal answers. Correspondence should be sent in care of COMPUTE!.





## VERIFLEX

Lifetime warranty. Free plastic library case. Boxed in tens.

	50+	100+
5 1/4" Single Sided/ Double Density.....	.99	.94
5 1/4" Double Sided/ Double Density.....	1.09	1.04

## Nashua

Boxed in tens.

	50+	100+
5 1/4" Single Sided/ Double Density.....	1.01	.96
5 1/4" Double Sided/ Double Density.....	1.11	1.06

## SONY

The original 3 1/2" micro, boxed in tens.

	20+	50+	100+
3 1/2" Single Sided	2.28	2.25	2.19
3 1/2" Double Sided	3.45	3.39	3.35

## DATA-CASE

Stores/files up to 50 5 1/4" disks. Includes dividers and labels.

Compare at only \$8.95

## RIBBONS

Sold in dozens. Each

Epson MX-70/80 .....	\$3.29
Epson MX-100 .....	4.65
Okidata Micro 801/82A/83A/92/93 .....	1.39
Okidata Micro 84/94 .....	3.35

Call for other models!

CALL  
TOLL FREE 1-800-USA-FLEX  
7 to 7 Central Standard Time  
In Illinois 1-312-351-9700

Please mention this ad when calling in your order. Master Card, Visa, C D D or for 30 days with approval credit on purchases over \$200.00. Shipping/Handling additional. Prices subject to change without notice.

# COMARK

COMARK, INC. 135 N. Brandon  
Glendale Heights, Illinois 60139

STARPOINT SOFTWARE proudly presents

# ISEPIC

■+■+■+■+■+■+■+■+■

[say Icepick], a revolutionary new concept in software de-protection for the Commodore 64. ISEPIC is not a disk duplication system, but an extraordinary hardware/software combination that actually bypasses any disk protection scheme. ISEPIC captures and saves the protected program as it runs in the 64's memory, this "snapshot" becomes accessible to the user for complete inspection and alteration. From this image, ISEPIC can automatically create a compact, auto-booting, fast-loading file which is completely unprotected and self contained.

- ☆ Copies ALL memory-resident software
- ☆ ISEPIC'd programs load many times faster than originals
- ☆ ISEPIC is invisible to software—cannot be defeated
- ☆ Eliminates drive "knock" due to antique protection schemes—adds years of life to your drive
- ☆ Automatically "cracks" protected programs into single, auto-booting, super-fast loading files
- ☆ Place multiple programs on a single diskette
- ☆ Create auto-booting, fast-loading versions of your own programs
- ☆ Cracked programs are completely self-contained and run independently of the ISEPIC adapter
- ☆ Copies software with a flick of a switch
- ☆ ISEPIC comes complete and ready-to-run, just plug into expansion port
- ☆ Programs cracked by ISEPIC may be used on MSD or 4040 drives as well as hard disks regardless of original protection schemes

When ordering by mail:

- \* \$64.95 + 3.00 shipping
- \* \$64.95 + 4.00 CDD orders
- \* Calif. residents add 6% sales tax
- \* VISA or Mastercard accepted
- \* Shipping out of USA \$6.00

Please allow 4-6 weeks for delivery.

WRITE OR PHONE  
**STARPOINT SOFTWARE**  
Star Point 10      Gazelle, CA 95034      [916] 435-2371



# On the Road With Fred D'Ignazio

Fred D'Ignazio, Associate Editor

## Buying The Right Educational Software

What types of educational software are people buying? What kind of software do they need?

In the opinion of many educators, the most important use of the computer as a learning tool lies in improving students' thinking skills through the use of programming languages like Logo; simulations and builder kits like *The Whatsit Corporation* (Sunburst) and *Operation: Frog* (Scholastic); microworlds like *Rocky's Boots* and *Robot Odyssey I* (The Learning Company); and problem-solving software like *The Pond*, *SemCalc*, *The Factory*, and *Geometric Supposer* (Sunburst).

But the sales of these products are dwarfed by the sales of drill and practice programs and learning games. A quick glance at a recent *Billboard* chart of the ten best-selling educational packages shows that eight of them are drill and practice programs and the remaining two are learning games. Of the drill and practice programs, two teach how to type, three teach basic math skills, two help students practice for the SAT college-entry exam, and one teaches basic vocabulary and spelling skills.

A look at TESS (The Educational Software Selector), published by the Educational Products Information Exchange and the Consumers Union, shows the same dominance of drill and practice programs. Of the 7,000 programs listed in TESS, almost 70 percent are drill and practice programs, and only 8.3 percent are simulation and problem-solving programs. (For more information about TESS, write to EPIE, P.O. Box 839, Water Mill, NY 11976.)

Most experts in educational computing have been critical of drill and practice programs for years. And most experts agree that

problem-solving and simulation software is the most challenging and interesting software for anyone learning on a computer. If this is true, why are companies producing so much drill and practice software? And, more importantly, why do people prefer it?

## Wary Adults

The answer is that most parents (and many teachers) are not ready for new kinds of software that teach new skills in new, unfamiliar ways. They don't understand how the programs work or what they're supposed to teach, or why it's important, and they don't see where the programs fit into their children's learning. And since they don't see a need for the programs, they don't buy them.

This is a natural reaction. For most people, computers are still a strange, almost alien, new medium. Many parents are still uncomfortable having a computer in their home. And many teachers, too, feel privately fearful of computers. They see the computer as a threat—a means to automate them out of a job. The more the computer's role in the classroom grows, the more they see their own role being eroded.

In addition, problem-solving and thinking-skill software is an unfamiliar, new application of computers. We have a new medium (computers) trying to teach new concepts (logic and thinking skills) using new methods (microworlds, simulations, etc.). This is too much novelty for the average consumer—whether that consumer is a parent or a school system. As a result, most consumers are buying drill and practice programs and learning games because at least this way they see the computer teaching practical, necessary, and familiar skills—using a nonthreatening, understandable approach. And in the

classroom, since the skills are familiar, the programs that teach them are more easily integrated into a teacher's lesson plan and curriculum. A program that teaches a child some spelling words can slip effortlessly into a curriculum, but what does a teacher do with a program that teaches a child how to think?

For the present, most parents will be buying and using drill and practice software and learning games, and ignoring problem-solving and simulation software. Does this mean that companies should stop producing these more challenging, yet less successful programs? Hardly. Instead, educators and software companies need to launch a major effort to communicate to parents and teachers the importance of the new kinds of software. To do this, software companies must demonstrate to parents and teachers why learning these skills is important, and how the software fits into their children's learning curriculum. ©

# For Commodore owners, the latest collections from COMPUTE! Books

## COMPUTE!'s Commodore Collection, Volume 1 Twenty-eight original programs for the VIC and 64 Edited

The *Commodore Collection*, an anthology of 28 previously unpublished programs for the VIC-20 and Commodore 64, continues the COMPUTE! tradition of practical programs in easy-to-use form. Includes exciting games, stimulating educational programs, valuable utilities, and useful applications to make your Commodore computer an exciting part of every day.

**\$12.95** ISBN 0-942386-55-8



## COMPUTE!'s Commodore Collection, Volume 2 Programs for the VIC and 64

Edited

Exciting games, sophisticated applications, versatile educational routines, and helpful programming aids for the VIC-20 and Commodore 64 highlight this second volume in COMPUTE!'s Commodore Collection series. Included are some of the best articles and programs from recent issues of *COMPUTE!* and *COMPUTE!'s Gazette*, as well as many programs published here for the first time. Designed for Commodore computer users of all levels, it's a book that every VIC or 64 owner will want to have.

**\$12.95** ISBN 0-942386-70-1

To charge your order, call toll free 800-334-0868 or send in this coupon with your payment to COMPUTE! Books, P.O. Box 5058, Greensboro, NC 27403.

Please add \$2.00 shipping and handling per copy ordered.

- ☐ Payment enclosed (check or money order)  
☐ Charge: ☐ Visa ☐ MasterCard ☐ American Express

Account No. \_\_\_\_\_ Exp. Date \_\_\_\_\_

Signature \_\_\_\_\_

Name \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Please allow 6-8 weeks for delivery

7681101

\_\_\_\_ copies of *COMPUTE!'s Commodore Collection, Volume 1* @ \$12.95 ea. \_\_\_\_\_

\_\_\_\_ copies of *COMPUTE!'s Commodore Collection, Volume 2* @ \$12.95 ea. \_\_\_\_\_

Shipping & handling \_\_\_\_\_

\$2.00/book \_\_\_\_\_

Total payment \_\_\_\_\_





## MONITORS

390 Green	\$129.00
390 Amber	\$130.00
310 Amber IBM Plug	\$160.00
Color 300 Composite	\$209.00
Color 500 Composite	\$309.00
Color 600 Hi-Res (960x400)	\$399.00
Color 700 Hi-Res (720x400)	\$499.00
Color 710 Long Phosphor	\$570.00

12" Amber/Green Composite	\$59.99
12" Amber/Green TTL (ea)	\$119.00

JB 1280 Green	\$59.99
JB 1201/1205 (ea)	\$90.99
JB 1270 Green	\$130.00
JB 1275 Amber	\$149.00
JB 1280 G TTL/1295 A TTL	\$149.00
JC 1400 Color	\$209.00
JC 1410 RGB	\$999.00

MAX-12E Amber	\$189.00
MX-9 "R" RGB	\$469.00
HX-9E Enhanced	\$519.00
HS-12 12" RGB	\$499.00
MX-12E Enhanced	\$669.00
SR-12 Hi-Res	\$599.00
SR-12 P Enhanced	\$649.00

115 12" Green Mono	\$119.00
116 12" Amber Mono	\$119.00
121 Green TTL	\$139.00
122 Amber TTL	\$149.00
400 Med-Res RGB	\$259.00
400 Hi-Res RGB (IBM)	\$429.00
420 Ultra Hi-Res RGB	\$599.00

\$400 Quadchrome I	\$479.00
\$410 Quadchrome II	\$499.00
\$420 Amberchrome	\$179.00
\$500 Dual Screen	\$1499.00

ZVM 1201/123	\$73.99
ZVM 124 IBM Amber	\$149.00
ZVM 130 Color	\$269.00
ZVM 131 Color	\$279.00
ZVM 133 RGB	\$429.00
ZVM 135 RGB/Color	\$459.00
ZVM 136 RGB/Color	\$599.00
1280, 1290, 1340	CALL

## INTERFACES

GraphicCard	\$79.99
Serial Card	\$99.99
Microbuffer II	\$109.00
Microbuffer 30K	\$189.00

MicroLink	from \$139.00
Elazer (Japan)	from \$79.99

Grappier CD (C64)	\$99.99
Grappier + Apple	\$89.99
Grappier 16K+ (Apple)	\$159.00

## DIGITAL DEVICES

Ape Face (Atari)	\$49.99
U-Print A (Atari)	\$54.99
U-16BBuffer (Atari)	\$74.99
U-Call Interface (Atari)	\$59.99
U-Print C (C64)	\$49.99
P-16 Print Buffer	\$79.99

MB1150 Parallel (Atari)	\$79.99
NP-1150 Parallel (Atari)	\$69.99
NP1150M (Atari 1050/5)	\$69.99
MicroBuffer 64K Port Buffer	\$109.00

## PRINTERS

AT-100 Atari Interface Printer	\$199.00
AT-550 Atari Dual Mode	\$249.00
GP-100 Parallel Interface	\$199.00
GP-700 Color Printer	\$449.00
GP-560 Parallel Printer	\$229.00
Elite S Letter Quality	\$226.00
Elite S C&I Interface	\$249.00
Perman 3-pin	\$289.00

MSP-10 (30 col)	\$349.00
MSP-15 (132 col)	\$499.00
MSP-20 (30 col)	\$499.00
MSP-25 (132 col)	\$679.00

Prowriter 7500	\$219.00
Prowriter 8510P	\$299.00
Prowriter 8510 NLQ	\$329.00
Prowriter 1550P	\$499.00
F10 40P Stylewriter	\$999.00
F10 55 Stylewriter	\$1049.00
Prowriter 8510-NLQ	\$329.00

ComWinlet Letter Quality	\$399.00
--------------------------	----------

Laser LP-300	\$2799.00
--------------	-----------

D35 Daywriter	\$599.00
430 109 Daywriter	\$1749.00
D301F Daywriter	CALL

2000	\$749.00
------	----------

EPSON	
RX-600, FX-400, LX-400, JX-400	CALL
FX-100+, RX-100, LX-100	CALL
HomeWriter 100	CALL
NEW! LX-400, SQ-2000, DX-100, DX-200	CALL

OKIDATA	
84 162 192, 193, 2410	CALL
Quante 10 (Specify C&I/MS/159)	CALL
Diamond 20 (IBM)	CALL

OLYMPIA	
Neodisplay Dot Matrix	\$299.00
Compact RD	\$339.00
Compact 2	\$369.00

Panasonic	
KX1080	\$109.00
KX1091	\$279.00
KX1092	\$109.00
KX1093	\$99.00

Quikset	\$359.00
---------	----------

SILVER-REED	
550 Letter Quality	\$279.00
550 Letter Quality	\$419.00
770 Letter Quality	\$759.00

S&S/S&S Series	CALL
Powertype Letter Quality	CALL

TOSHIBA	
1340 (30 column)	\$599.00
P35P (132 column)	\$1299.00

## PC COMPATIBLES

## IBM PC SYSTEMS

Configured to your specification.  
Call for Best Price!  
IBM-PC, IBM-PC II, IBM-XT, IBM-AT

2220 Dual Portable	\$1499.00
4220 Dual Desktop	\$1399.00
4620 10 meg Desktop	\$2299.00

Safari (7900)	CALL
6300	CALL

## SOFTWARE FOR IBM

Electronic Desk	\$199.00
ASHTON-TATE	
Fontworks	\$399.00
dBase II	\$299.00
dBase III	\$299.00

Turbo Pascal 3.0	\$49.99
Sidekick	\$39.99

Get Organized	\$89.00
Get in Shape	\$39.99
Music Manager	\$59.99
One-on-One	\$29.99
Financial Cookbook	\$39.99

Harvard Software Inc.	
Harvard Project Manager	\$209.00
Total Project Manager	\$269.00
Human Edge™	
Communication Edge	\$19.00
Management Edge	\$119.00
Navigation Edge	\$139.00
Sales Edge	\$119.00

INSI	\$84.99
Lotus	
Symphony	\$439.00
1-2-3	\$209.00

WordStar 2000	\$249.00
WordStar 2000+	\$319.00
Flight Simulator	\$39.99
MultiPlan	\$129.00

Crossark	\$99.99
MicroSoft	
R Base 4000	\$249.00
R Base 5000	\$249.00
Clout 20	\$199.00

MultiMate	\$249.00
MultiMate	
PeachPack (QUADRA)	\$159.00

Access (NEW)	\$70.00
Write/Graph/File/Plan	(ea) \$79.99
Report	\$74.99
Proof	\$59.99
Mac Software	CALL

Professional Software	
PC RunThe Bus	\$249.00
Enable	\$99.00

SORIUMUS	
APR/AR/AR/AR	(ea) \$295.00
Supercalc II	\$195.00
EasyWriter II System	\$195.00
Super Project	\$195.00

Open Access	\$379.00
SSI Software	
Word Perfect	\$299.00
File Manager (IBM)	\$339.00

THOUGHTWARE	
Trigger	\$269.00
Self Self Self	
Training	\$299.00
Application	\$179.00

PC-151-01 Single Desktop	CALL
PC-151-02 Dual Desktop	CALL
PC-151-03 10 meg Desktop	CALL
PC-151-01 Single Portable	CALL
PC-151-02 Dual Portable	CALL
2-200 (AT)	CALL
171 (Portable)	CALL
138 (Transposable)	CALL
148 (DeskTop)	CALL

corona	
PPC400 Dual Portable	\$1799.00
PPC420 10 meg Portable	\$2399.00
PPC4022 10 meg Desktop	\$1999.00

SANYO	
MBC 500-2 Single Desk	\$999.00
MBC 550-2 Dual Desk	\$999.00
MBC 770 Portable	\$1799.00
MBC 511 10 meg	CALL
MBC 675 Portable	CALL
MBC 800 Desktop	CALL

## MULTIFUNCTION CARDS

AST	
Star Pack Plus	\$239.00
Mega Plus II	\$269.00
10 Plus II	\$159.00
Advantage AT	\$799.00
Graph Pak	\$399.00
Monograph Plus	\$299.00
Preview Mono	\$299.00
PC Net Cards	\$379.00
251111 Grapner	\$799.00
6251112 Remote	\$579.00
3780 Emulation Card	\$699.00
95C Bayline	\$499.00

dca	
IRMA 3270	\$579.00
IRMA Print	\$99.00

eagle	
Memo Card (Closeout)	\$99.99

EVEREX	
Color Card (Graphics Edge)	\$269.00
Magic Card	\$199.00

HERCULES	
Graphics	\$299.00
Color	\$159.00

Associates	
IDEAmax - ZPR, 64K, C, S, P	\$229.00
IDEAmini - YPR, C, S, P	\$189.00
IDEAmax - MPR 128K	\$229.00
IDEAshare Software	\$219.00
IDEA 5251	\$99.00

MYLEX	
The Charm	\$499.00

PARADISE	
Modular Graphics Card	\$274.00
Multi Display Card	\$289.00
Flex Pack C, S	\$159.00

PLANTRONICS	
Color Plus	\$399.00

QUADRAM	
Quadram AT	\$119.00
Quadram - AT (128K)	\$249.00
The Gold Quadboard	\$449.00
The Silver Quadboard	\$399.00
Expanded Quadboard	\$219.00
Quad 512+	\$299.00
Liberty	\$399.00
QuadPrint	\$499.00
Dual 1K	\$299.00
Quadcolor I	\$199.00
Quadcolor Expansion Chassis	\$499.00
Quadcolor Chassis Memory	\$199.00
Quadcolor Jr	\$199.00
Chronograph	\$79.99
Parallel Interface Board	\$99.99

1-800-233-8950

COMPUTER MAIL ORDER

## Program Your Own EPROMS

► VIC 20 **\$99.50**  
► C 64

PLUGS INTO USER PORT.  
NOTHING ELSE NEEDED.  
EASY TO USE. VERSATILE.

- Read or Program. One byte or 32K bytes!

OR Use like a disk drive. LOAD,  
SAVE, GET, INPUT, PRINT, CMD,  
OPEN, CLOSE—EPROM FILES!

Our software lets you use familiar BASIC commands to create, modify, scratch files on readily available EPROM chips. Adds a new dimension to your computing capability. Works with most ML Monitors too.

- Make Auto-Start Cartridges of your programs
- The promenade™ C1 gives you 4 programming voltages, 2 EPROM supply voltages, 3 intelligent programming algorithms, 15 bit chip addressing, 3 LED's and NO switches. Your computer controls everything from software!
- Testtool socket. Anti-static aluminum housing
- EPROMS, cartridge PC boards, etc. at extra charge.
- Some EPROM types you can use with the promenade™

2716	2632	480232P	27130	5733	22815A*
2416	1703	2704	27236	6145	52817
2718	27232	2704	66764	2812*	48018P*
27C16	27C32A	27C84	66766	2812*	

\* Commodore Business Machines

\* General electrically erasable types

Call Toll Free 800-421-7731

In California: 800-421-7748



**JASON-RANHEIM**

680 Parrott St., San Jose, CA 95112



## SwiftCalc — spreadsheet superiority!

**NOW WITH  
SIDEWAYS!**



A powerful,  
easy-to-use  
electronic  
spreadsheet  
for home and  
small business  
use

For  
Commodore 64  
and  
IBM  
personal  
computers

444 Lake Cook Rd.  
Deerfield, IL 60015  
1-800-323-9755

# TIMEWORKS

More power for your dollar.

## HOTWARE: Software Best Sellers

						Systems				
This Month	Last Month	Title	Publisher	Remarks		Apple	Atari	Commodore	IBM	Macintosh
<b>Entertainment</b>										
1.	3.	<i>Flight Simulator II</i>	SubLogic	Aircraft simulation		•	•	•		
2.	2.	<i>The Hitchhiker's Guide to The Galaxy</i>	Infocom	Comic adventure strategy game		•	•	•	•	•
3.	1.	<i>Flight Simulator</i>	Microsoft	Aircraft simulation					•	
4.	4.	<i>F-15 Strike Eagle</i>	MicroProse	Air combat simulation		•	•	•		
5.		<i>Summer Games II</i>	Epyx	Action sports game		•	•	•	•	
<b>Education</b>										
1.	3.	<i>New Improved MasterType</i>	Scarborough	Typing instruction program		•	•	•	•	•
2.	1.	<i>Typing Tutor III</i>	Simon & Schuster	Typing instruction program		•	•	•	•	•
3.	2.	<i>Math Blaster!</i>	Davidson	Introductory math program, ages 6-12		•	•	•	•	
4.	5.	<i>Early Games</i>	Springboard	Educational games, ages 2-6		•	•	•	•	
5.	4.	<i>Word Attack!</i>	Davidson	Word meaning and usage program, grades 4-12		•	•	•	•	
<b>Home Management</b>										
1.	1.	<i>Print Shop</i>	Broderbund	Do-it-yourself print shop		•	•	•		
2.	2.	<i>Print Shop Graphics Library</i>	Broderbund	100 additional graphics		•	•	•		
3.	5.	<i>Bank Street Writer</i>	Broderbund	Word Processor		•	•	•	•	
4.		<i>HomePak</i>	Batteries Included	Word processing, telecommunications, & data management		•	•	•	•	
5.	3.	<i>Andrew Tablas's Managing Your Money</i>	Meca	Home financial program					•	

Copyright 1985 by Billboard Publications, Inc. Compiled by the Billboard Research Department and reprinted by permission. Data as of 6/66/85



# Telecomputing Today

Arlan R. Levitan

## SIGs: Behind The Scenes

It appears that 1985 is rapidly becoming a banner year for electronic Special Interest Groups (SIGs). Unfortunately, rapid growth is not without its problems.

Electronic SIGs (also known as discussion forums) are like computerized clubs where people with a common interest get together online to exchange information, ideas, comments, complaints, and public domain computer programs. Usually they are operated by commercial information services such as CompuServe. Besides computer-oriented SIGs, there are groups for doctors, lawyers, educators, and other professionals, not to mention enthusiasts of travel, cooking, literature, rock 'n' roll, politics, human sexuality, skiing, and ham radio. There are SIGs for every taste, and new ones are added every day.

SIGs are usually divided into three areas. The most active area is a message system subdivided into several sections by topic where users can read and leave messages. The oldest messages are overwritten when a new message is added, so each message tends to have a life of a week or less.

Each topic section also has a file access database where important messages may be stored more permanently as text files, and where users leave free copies of public domain software.

Finally, each SIG also has a conference area so online users can exchange messages in an immediate mode. Everything you type appears on the screens of others participating in the conference.

## Online Bucks

SIGs generate extra income for information services because they encourage more online usage. But how valuable are SIGs to an information service's bottom line?

Frankly, most SIGs generate only a modest amount of revenue, and some don't break even. There are, however, a significant number of SIGs that are monstrous hits, largely due to the work of their sysops (system operators). In fact, each of the top SIGs generates well over a million dollars a year! That can mean big bucks for the head sysop. It's not uncommon to find the sysop (or company sponsoring the SIG) pulling down at least 5 percent of the gross—\$50,000 a year or more. The assistant sysops generally work for "fame and glory" and get free time on the SIG for their efforts.

Although SIGs used to be the sole province of CompuServe, the revenues that they've been pulling in lately have prompted Delphi, The Source, and other services to get into the act. And the competition is just beginning to heat up. The lure of substantial amounts of SIG-generated money can do funny things to people.

For example, the entire staff of the Commodore SIG on one service recently jumped ship and set up shop with a competitor. The defection wasn't a secret—the sysops used the SIG message base to let members know where they had gone. And within the last year, the head sysop of another popular SIG had to contend with a palace coup staged by his assistant sysops. After the smoke cleared, there were two SIGs instead of one—a SIG headed by the original sysop and another by the rebels. Keeping the SIG peace might soon be a job for the United Nations.

Fortunately, most SIGs are very well run and perform a valuable function, acting as information clearinghouses. But as useful as they may be, there's no point in paying through the nose to enjoy the benefits of regular involvement

in a SIG. Here are some general tips that can help you save money when participating in a SIG.

## Be Selective

Don't try to read every new item in the message area. Many of the more popular SIGs can turn over a thousand messages or more in three or four days. Pick one or two sections with topics that appeal to you and stick with them.

Many SIGs let you read streams of messages without pausing for a response from you between items. To take full advantage of this feature, download the messages you want and read them offline. If you wish to reply to a message or leave a new one of your own, write your text offline as well; you can send the entire message in a matter of seconds rather than pecking away online. (See last month's column for hints on this technique.)

Unless you have a burning question that can't be answered by the folks who access the message section, pass up regularly attending special online conferences. Complete transcripts of the conferences are usually available in the file access areas shortly afterward. Given the speed that most people type, the text of a two-hour conference can usually be downloaded in about five minutes. If you can't resist conferences, don't bother to sign on at 1200 or 2400 bps if the service charges hourly premiums for these higher speeds. Ol' 300 baud is just fine for conferences and will keep the hourly rate you pay at a minimum.

Finally, don't ignore using the public domain program library of your local computer user group as an alternative to downloading files from SIGs. An entire diskful of programs from a user group usually costs no more than an hour of time on a SIG. ☐



## Atari Input/Output

Much of what I'm about to discuss this month has appeared in this column before. And the bulk of this information can also be found in the *Atari Technical Reference Manual*—presuming you can read "techlish." But this intro is necessary so we can start talking about the meat of our subject next month.

Still with me? Let's go. Atari's operating system (OS)—which, like the OS in any eight-bit computer, takes up the bulk of Read Only Memory (ROM)—is really a thing of beauty. In fact, it may be the only *consistent* OS to be found in any microcomputer, short of those sporting UNIX or its derivatives. CP/M and MS-DOS are such kludges that most commercial programs bypass the OS. (That's why there are so many "almost PC-compatibles.") The Commodore 64's operating system comes close, but its disk input/output is difficult at best. And Apple's ProDOS manual states that "users desiring to perform I/O to devices other than the disk drive are on their own!"

Atari users, on the other hand, enjoy a system with such complete support that, for most programs, all necessary input/output operations can be executed by calling a single subroutine! That subroutine is called, appropriately, *Central Input/Output* (CIO). By calling CIO with the proper values in certain memory locations and the proper pointer in the 6502's X register, your programs can perform such diverse operations as formatting a disk, drawing a line on the graphics screen, fetching a keystroke from the keyboard, sending output to the printer, or reading 25,000 bytes from a disk file.

Yet, CIO is invisible to most Atari users. For example, many of the capabilities which magazine and newsletter articles attribute to

BASIC are not part of BASIC at all. None of the graphics (including the so-called BASIC graphics modes) in Atari BASIC are actually performed by BASIC. Instead, BASIC simply translates the graphics command into a call to CIO. Atari developed this system in 1978, and it wasn't until the Macintosh appeared that such a revolutionary concept was repeated in a popular computer.

Generally, you have to become a machine language programmer to appreciate and use all the features of CIO. So why read any of this, then? Because calls to CIO can't perform *every* input/output task possible on Atari computers. What can't CIO do? First, let's take a glance at what it can do.

## Calling CIO

When CIO is called by a program, it expects the X register to contain a pointer to an *Input/Output Control Block* (IOCB). IOCBs are blocks of memory 16 bytes long which control CIO functions. The pointer value for the X register is easily calculated—it's actually the BASIC file number (as in OPEN #14, 0,"K:") multiplied times 16, because there are 16 bytes per IOCB. One of the bytes within the IOCB then tells CIO what function the program is requesting.

There are seven fundamental functions available: OPEN, CLOSE, STATUS, PRINT, INPUT, Block PUT, and Block GET. In addition, there are some *extended functions*. BASIC programmers are familiar with these because of the XIO statement, which allows you to call the functions from BASIC. But several other BASIC statements (including NOTE, POINT, DRAWTO, and LOCATE) access the CIO extended functions, too.

After CIO examines the IOCB and determines which function is being requested, it decides which

device (keyboard, disk, screen, etc.) should service the request. Then it calls an appropriate routine within the *device driver* for that device. (For example, the Disk Operating System—or more properly, the File Management System—is the device driver for the disk drive.) If the request is for an extended function, it is passed on unchanged to the device driver.

Well, with 256 possible command values, you would think that there isn't any request, however bizarre, which couldn't be serviced via CIO. In theory, true. In reality, you have to stop adding functions somewhere or you run out of memory. Thus Atari's CIO-based graphics have no function for drawing a circle, and DOS provides no command to format a disk without also writing a boot and directory.

If you want to draw a circle, you can write a routine to calculate and PLOT points or change screen memory directly. If you want to mess with the disk drive, though, you have to learn about another routine within the Atari ROMs, *Serial Input/Output*.

## The Mysterious SIO

SIO—which lets Atari computers talk to devices (such as printers and disk drives) which hook up to the serial bus—has acquired an undeserved aura of mystery. Actually, though, in some ways it is easier to call SIO than it is to call CIO!

For example, there is only one SIO "device" and only one Device Control Block (DCB). So even the X register pointer required by CIO isn't necessary when calling SIO. Intrigued? I hope so, because it's time to sign off for now. But after this brief overview, we're ready for next month's column: We'll show how to write a program to call SIO.



# IBM Personal Computing

Donald B. Trivette

## New Life For Aging PCs

This month's *COMPUTE!* is full of new wares from the Summer Consumer Electronics Show. But most of us can't afford to toss out the old and bring in the new—at least not yearly—so let's look at an interesting alternative.

One way to give new life to an aging PC is with a PCTurbo board by Orchid Technology (47790 Westinghouse Drive, Fremont, CA 94539). PCTurbo, which installs in one of the PC's expansion slots, contains a whole new computer and a whole new memory. It's like having a new computer inside an old computer. The old computer is the original IBM PC with its memory and Intel 8088 microprocessor. The new computer is completely contained on the PCTurbo board; it has an Intel 80186 microprocessor with up to 640K RAM.

You don't lose your old PC in the bargain—a PC with PCTurbo is like two separate computers. In PC mode, the computer runs just as it always did. But in PCTurbo mode, the machine zips through most programs three to four times faster. Two new DOS-like commands—TURBO and GOPC—switch from one mode to another.

The PCTurbo board executes all the usual program instructions, but handles input/output in a special way. For example, when a disk operation generates an input/output interrupt, the PCTurbo's 80186 processor passes the I/O task along to the slower 8088 processor. That means the 80186 can run a program at full speed without stopping to process time-consuming interruptions from the keyboard, screen, disk, and printer.

Nor is the PC's original memory left idle. Software supplied with PCTurbo allows that memory to be used as one or more electronic RAM disks and as a disk cache (pro-

nounced like "cash"). Disk caching has been around for years as a way to speed up I/O on mainframe computers. No matter how fast disk drives are, memory is three to four times faster. The PCTurbo disk-caching software automatically moves blocks of data (called sectors) from the disk to the cache (the old PC memory) so the data will be there when needed. In fact, the caching software anticipates requests: It moves the *next* disk sector into memory as well. All this is completely transparent to the application software—it still thinks it's reading and writing data on a floppy disk.

## A Dramatic Speed-Up

How well does disk-caching work? To find out, I plugged a PCTurbo board into my three-year-old 320K PC and invited over a friend.

Henry and his wife own an interior design firm; they use their PC with *Lotus 1-2-3* to keep track of chairs, tables, wallpaper, fabrics, taxes, and all the other charges for a job. I had heard Henry complaining about the spreadsheet for a modest residence he is doing for a surgeon. The spreadsheet is 28 columns wide by 389 rows deep—about 170,000 opulent bytes. When something is added—say a Ming vase—it takes the program 15.5 seconds to recalculate. (We all should have to spend a half-million dollars at four items per minute!) With the computer in PCTurbo mode, the recalculations were cut to six seconds. That's a speed increase of 250 percent (or six more items per minute).

My own problems more often involve checking the spelling in a manuscript rather than spending money more rapidly. I recalled a 2,049-word article (it took the PC three seconds to count the number of words; PCTurbo, one second) and set *Word Proof*, the IBM spelling

checker, to work. In 41 seconds, PCTurbo verified that the article had no errors; in regular PC mode, the job took 107 seconds. To speed things up even more, I moved *Word Proof's* 125,000-word dictionary from a floppy disk to an electronic RAM disk created in the regular PC memory. PCTurbo polished off all 2,049 words in 18 seconds (almost 114 words per second); the PC by itself took 77 seconds—more than four times longer.

And how much does it cost to find spelling errors faster than Superman? The invoice from Orchid Technology is \$747 for the PCTurbo board with 256K of memory, and \$375 for the optional 384K memory module to reach a total 640K. The fully loaded board—PCTurbo with 640K—costs \$1,122. Not bad for an accessory that almost turns a three-year-old PC into a PC AT.

Almost, that is, because the PC won't do a few things in PCTurbo mode. IBM BASIC is an IBM proprietary product built into a chip on the PC's main circuit board; it's not available to PCTurbo, so you must switch to PC mode to run BASIC or BASICA. But compiled BASIC runs just fine with PCTurbo. Also, the PCTurbo board I tested wouldn't run communications programs (it couldn't find the modem). However, Orchid says it has a solution to that in the works. Somehow it's comforting to know there's lots of life left in the old PC.

## Odds And Ends

The second edition of IBM's *Directory of Personally Developed Software* has been out for several months. To get a free copy, call 1-800-IBM-PCSW (a.k.a. 1-800-426-7279). The new directory lists 58 software products ranging in price from \$15 to \$150, although most cost about \$20. There are some real bargains here. ©



# Programming the TI

C. Regina

## Trivia Quiz

There has been quite a big fuss about the trivia-type question and answer games—Trivial Pursuit and the various takeoffs. I'm certain that you've also seen ads for computer versions of these board games. Have you ever wanted to buy one? Wait! Write your own. This month's program, "Trivia Quiz," gives you the basic structure for a question and answer game.

The main part of either the board games or the computer versions is the bank of questions. Of course, we can't publish a program with questions already included because it would take up the whole magazine. To make your own game, you must invent your own questions. Teachers can use this program for a basic essay-type quiz on any subject, and families can think up questions that appeal to their interests.

The computer is used to shuffle the questions, or to randomly choose one question at a time and give its corresponding answer. Once a question comes up, it is not used again.

I've previously published programs for a matching quiz and a multiple-choice quiz. This program creates a fill-in-the-answer or essay quiz. You can even print out copies of tests that have the questions in a different order for each person.

## Modifying The Program

In each of the program's DATA statements, first write a question, then type a comma, then put the answer. Remember to adjust the spacing so it looks nice on the screen (no words split across screen boundaries). If you need commas within the question or the answer, you'll need to use quotes. Just to give you an idea, the sample program contains questions from nursery rhymes and fairy tales.

Adjust the DIMension statement for the number of questions you put in your DATA statements. For example, I've used only 20 questions. You'll probably want many more for a trivia quiz. Also define N in line 120 for the number of possible questions.

After the computer chooses and prints a question, the player inputs the answer. The computer then prints the correct answer. The computer does not keep score, however, because you may want to accept several forms of the answer. For instance, you could accept the answer to the question of what Little Miss Muffet ate as "curds and whey," "her curds and whey," or even "cottage cheese"—but you would not accept "porridge."

Trivia Quiz simply asks the questions, so any number of people can play. You may want to expand the program to ask questions of each player in turn, to add scoring, or to make a more complex game. Of course, you can add graphics and sound, too.

If you wish to save typing effort, send a blank cassette or disk, a stamped, self-addressed mailer, and \$3 to:

C. Regina  
P.O. Box 1502  
Cedar City, UT 84720

Please be sure to specify that you want the TI version of Trivia Quiz.

## Answers To Reader Questions

I cover TI BASIC exclusively in this column because most COMPUTE! readers have console BASIC only. For other programming languages and hardware information, a good newsletter is published by the 99'ers User Group Association, 3535 South H Street, #93, Bakersfield, CA 93304. This is a nonprofit organization that communicates

with hundreds of local user groups.

User groups are still going strong and are your best source of information and help. Many publish excellent newsletters (sorry, too numerous to mention here, and I wouldn't want to slight anyone).

New software is still being produced for the TI—I guess because there are over two million TIs out there. Most new programs are entertainment and educational titles. And yes, the Extended BASIC module remains available.

Hardware also is abundant. There are separate units for each peripheral, which saves you money if you need only one item. There are also combinations. One attractive unit I've used is CorComp's expansion unit that contains an RS-232 interface, 32K memory expansion, and a disk controller, all in a box about six inches wide that connects to the side of the console. It works just like the TI Peripheral Expansion Box but without the bulk and the noisy fan. My only complaint is that the disk drive connector isn't the same as the TI disk drive ribbon cable, but I understand Radio Shack has the necessary connections.

Next month I'll discuss how to use peripherals in your programming.

## Trivia Quiz

```
100 REM TRIVIA QUIZ
110 DIM S$(20),A$(20)
120 N=20
130 CALL CLEAR
140 PRINT TAB(8);"TRIVIA QUIZ"
150 PRINT :;"A QUESTION WILL BE SHOWN."
160 PRINT :;"TYPE THE ANSWER (WITHOUT)"
170 PRINT :;"COMMAS) THEN PRESS ENTER."
180 PRINT :;"THE CORRECT ANSWER IS SHOWN."
190 PRINT :;"PRESS THE SPACE BAR TO"
200 PRINT :;"CONTINUE."
210 FOR C=1 TO N
220 READ S$(C),A$(C)
230 NEXT C
```

```

240 PRINT :;"PRESS ANY KEY
    TO START."
250 CALL KEY(0,K,B)
260 IF S<1 THEN 250
270 FOR C=1 TO N
280 CALL CLEAR
290 RANDOMIZE
300 R=INT(N*RND)+1
310 IF S$(R)="" THEN 300
320 PRINT S$(R):;
330 CALL SOUND(100,1497,2)
340 INPUT B$
350 PRINT :A$(R)
360 CALL KEY(0,K,B)
370 IF K<>32 THEN 360
380 S$(R)=""
390 NEXT C
400 CALL CLEAR
410 REM PUT QUESTIONS HERE

```

```

420 DATA WHO LOST THEIR MIT
    TENS?,THE THREE LITTLE
    KITTENS
430 DATA WHO WERE THE THREE
    MEN IN A TUB?,"THE BUT
    CHER, THE BAKER, THE CA
    NOLESTICK MAKER"
440 DATA WHEN WILL THE CRAO
    LE ROCK?,WHEN THE WIND
    BLOWS
450 DATA WHAT DID JACK AND
    JILL(4 SPACES)FETCH?,A
    PAIL OF WATER
460 DATA WHO CUT OFF MICE'S
    TAILS?,THE FARMER'S WI
    FE
470 DATA HOW MAY BAGS OF WO
    OL DID THEBLACK SHEEP H
    AVE?,THREE
480 DATA WHO VISITED THE TH
    REE BEARS?HOME?,GOLOILO
    CKB
490 DATA WHO CHOKED ON AN A
    PPLE?,SNOW WHITE
500 DATA WHO SANG FOR HIS B
    UPPER?,LITTLE TOMMY TUC
    KER
510 DATA WHAT DID THE THREE
    PIGS USE TO BUILD THEI
    R HOUSES?,"STRAW, STICK
    S, BRICKS"
520 DATA WHO USED HER LONG
    HAIR TO(3 SPACES)SEE TH
    E PRINCE?,RAPUNZEL
530 DATA WHAT DID JACK FIND
    IN HIS(3 SPACES)CHRIST
    MAS PIE?,A PLUM
540 DATA WHAT DID LITTLE MI
    SS MUFFET EAT?,CURDS AN
    D WHEY
550 DATA WHO WAS LITTLE RED
    RIDING(3 SPACES)HOOD B
    OING TO VISIT?,HER GRAN
    DMOTHER
560 DATA WHAT COULDN'T JACK
    SPRAT(4 SPACES)EAT?,FA
    T
570 DATA WHO STOLE A PIG AN
    D AWAY DORUN?,"TOM, TH
    E PIPER'S SON"
580 DATA WHO JUMPED OVER A
    (11 SPACES)CANOLESTICK?,
    JACK
590 DATA WHO SLEPT WITH HIS
    STOCKINSON?,HY SON JO
    HN
600 DATA WHOSE FOOT FIT THE
    GLASS(4 SPACES)SLIPPER
    ?,CINDERELLA
610 DATA WHAT KIND OF MEAT
    DID ONE OFTHE LITTLE PI
    GGIES HAVE?,ROAST BEEF
620 END

```

## TI Webster Dines Out

In line 480 of this game program from the June issue (Program 6, p. 57), the DISPLAY AT(3,22) should be DISPLAY AT(23,22). Reader Andrew Sonon supplied this correction, which moves the score indication to its proper place on the screen.

## Apple MLX Error Messages

Although there are no errors in the "Apple MLX" listing from the June issue (p. 114), a number of readers have encountered DISK ERROR messages at unusual times. During normal operation of the program, the only errors that should occur are ones involving disk access; line 100 traps these errors. However, a side effect of this error trapping is that typing mistakes you make while entering MLX can also produce the message in line 610. Thus, if MLX gives you a DISK ERROR message when you are not accessing the disk, you need to check for a typing mistake in the MLX program. Lines 330-340 are a common problem area. Make sure you have not confused the letter O (used in the variable names OS and O in those lines) with the number zero, which also appears in line 340. In COMPUTE's listings, a zero always has a diagonal slash through it.

## Commodore Disk Editor

The POKE 1024, PEEK(254) in line 260 of this disk utility program from the June issue (p. 98) prevents you from changing the value of the first byte in a sector. This is the track number for the next sector in the file, so you may not need to change its value very often. However, you can modify the program to allow editing of the first byte by replacing the GOTO 260 at the end of line 310 with WAIT 198,255: GOTO 280.

## Editing Enhancement For Softball Statistics

This record-keeping program from the July issue (p. 30) works as published for all the listed computers. However, the data input process can be simplified by allowing corrections for each player's statistics. To accomplish this, make the following changes to the general program (Program 1):

```

545 PRINT
546 PRINT "EVERYTHING OK (Y/N)
    ?"
547 INPUT A$
548 IF A$<>"N" AND A$<>"Y" THEN
    G 545
549 IF A$="N" THEN 420

```

TI-99/4A users should also make the following additional changes:

```

548 IF (A$<>"N")*(A$<>"Y") THEN
    G 545

```

Atari users should also make the following additional changes:

```

250 NEXT I:PRINT "(DOWN)E
    verything OK (Y/N)?":
    GOSUB 630:IF A$="N" T
    HEN 210
255 GOSUB 640

```

## Mindbusters For Atari DOS 3.0

To use the Atari version of this thinking game from the April issue (p. 50) with DOS 3.0, you must delete the DIM K(255) in line 5 and make the following changes to line 1:

```

1 DIM K(255):FOR I = 0 TO 255: K(I) = 0:
    NEXT I: GOTO 5

```

## Archon II: Adept

Arthur Leyenberger

*Requirements: Commodore 64 or 128; Atari 400/800 or XL with at least 48K RAM; or an Apple II-series computer with at least 48K RAM. All versions also require a disk drive and a joystick.*

In any entertainment business, whether it is movies, books, or videogames, there is a natural tendency to produce sequels to existing hits. Making sequels can be approached in a number of ways. Often the sequel is just more of the same thing, such as *Jaws II* and *Jaws 3-D*; the hope is that there will be continued demand for more of the same thing. The risk in this approach is that people may eventually grow tired of the old formula.

Another approach is to use the same basic theme of the original, but add something new or better to the sequel—as in the successful *Star Wars* and *Star Trek* films. This is also the case with *Archon II: Adept*, a new game from Free Fall Associates, published by Electronic Arts.

*Adept* was designed to be a game that has the same mixture of strategy, action, and play mechanics as the original *Archon*. But it is sufficiently different to please both new players and long-time *Archon* devotees.

### Casting Magic Spells

*Adept* is basically a game of magic, focusing on energy and resource management. Unlike *Archon*, it allows people of different skill levels to compete more equally. The combination of strategic board play and individual combat means that people who don't have fantastic reflexes have a reasonable chance of winning.

Each side starts with four Adepts, similar to the Wizard and the Sorceress pieces in *Archon*. The game begins with one Adept on each of the four elements: Earth, Water, Air, and Fire. The elements are represented by colored bands on the screen. Your pieces are shown

vertically on each side of the screen, with the more powerful, flexible pieces at the top.

The more powerful pieces require more energy to manipulate. Choosing the beginner level gives you the most energy while choosing the advanced level gives you the least. The play level also affects the speed of the pieces in combat. Each piece can cast spells, and every spell costs magical energy. You can cast as many spells as you want with any piece as long as you can afford it. To gain magical energy, you must occupy "power points." If you occupy all of the power points, you win the game.

You shape your magical army depending on such factors as your skill with a particular piece, a certain strategy, or the pieces you like. Each side has four different elemental pieces that can be called upon. They are strongest in their own element but can be played in any element. Some of the pieces are common to both sides, while others are unique.

When you choose a piece, you are shown the amount of energy you currently have as well as how much energy it costs to use that piece or cast a spell. Although many of the spells are familiar to the experienced *Archon* player, some have new twists. For example, the Imprison spell lets you trap an enemy piece as long as you have the energy to pay for it. Casting and maintaining this spell costs energy during every turn, so imprisoning everyone in sight would soon drain your resources and lose you the game.

### Apocalypse Now

All of the spells can be cast repeatedly, with one exception: the Apocalypse. You can cast this spell only once, since it begins a battle that ends the game. It is a one-on-one, winner-take-all battle that is shaped by your strategic position: the amount of energy, number of pieces, and surviving Adepts you have left. The result can be either a well-matched or very one-sided battle.

This go-for-broke spell typically is cast in two situations. One is if you are way ahead and, for some reason, are having trouble occupying the last power

point. The other situation is when you're in danger of losing the game and have no other way out. Since the Apocalypse spell is expensive, casting it in a weak position weakens you still more.

*Adept* contains a wider variety of creatures than are found in *Archon*. Each piece has a unique weapon and performs best in a certain element. For example, the Juggernaut is best suited to Air and uses itself as a missile. When fired, it glows and charges across the screen. While in motion, it is invulnerable and destroys anything in its path. Using the Juggernaut in the water, however, significantly slows it down and makes it more vulnerable to attack.

In 1983, *Archon* ranked at the top of almost every gamer's list. It still belongs in the videogame hall of fame. *Archon II: Adept* is even better than *Archon* and should rank even higher with experienced *Archon* devotees as well as with players new to the world of magic and strategy.

*Archon II: Adept*  
Electronic Arts  
2755 Campus Drive  
San Mateo, CA 94403  
\$33 (Commodore & Atari)  
\$40 (Apple)

## WordPerfect For IBM

Richard Mansfield, Senior Editor

*Requirements: IBM PC or compatible with at least 192K of RAM, two disk drives, DOS 2.0 or higher, and a printer. Not compatible with the PCjr.*

This is an excellent word processor. In addition to performing all the usual tasks with speed and efficiency, *WordPerfect* includes many features which are either rare or unique.

Perhaps its best single feature is that it gives you a blank screen to write on—no distractions from control codes, command lines, or other clutter. Just a couple of unobtrusive numbers in the lower-right corner to identify the col-



# DISK WORLD!

## is pleased to announce the lowest-prices ever on brand name diskettes!

You've seen ads for lower priced diskettes.

But did you ever notice that they usually don't carry or list a brand name. There's a reason for that: there is no brand name to be listed, since these low-cost "generic" diskettes are positively called garbage within the diskette industry.

One or two individuals buy reeled product from lower-quality manufacturers and put it out in the market place without a label...and without any meaningful guarantee that you're getting good product.

But not at DISK WORLD!

We deal only with branded merchandise...with diskettes made by people who are proud to have their name known.

**3M:** Almost every diskette you buy has a little 3M in it. 3M manufactures more blank diskette media than anyone else...and probably produces more computer media in all forms than any other manufacturer. This is factory-fresh 3M product, boxed in 10's, reinforced hubs, user ID labels, write-protect tabs and a 100% LIFETIME WARRANTY.

5.25" 8DDO w/FREE Flo'n File 15.....	\$1.42 ea.
5.25" 8DDO w/FREE Flo'n File 15.....	\$1.74 ea.
(Above FREE offers subject to limited availability)	
5.25" 8DDO-2ETP.....	\$2.50 ea.
5.25" 8DDO-2ETP.....	\$2.57 ea.
5.25" 8DDO-HD for IBM PC/AT.....	\$3.93 ea.
5.25" 8DDO-135TPH for Mac and others.....	\$2.96 ea.

**BASF:** Exceptional value from one of the world's largest companies. Except where indicated, BASF diskettes are packaged in cardboard boxes of 10, reinforced hubs. Type sleeves, user ID labels and write-protect labels. Of course, there's a 100% LIFETIME WARRANTY on all BASF product.

5.25" 8DDO.....	\$1.14 ea.
5.25" 8DDO.....	\$1.31 ea.
5.25" 8DDO-HD 15/100PCAT.....	\$3.14 ea.
5.25" 8DDO in Bulk Packs of 50.....	.65 ea.
5.25" 8DDO in Bulk Packs of 50.....	\$1.12 ea.
5.25" 1-LIPPT diskettes in Bulk Packs of 50.....	\$1.20 ea.
5.25" 1-LIPPT diskettes in Bulk Packs of 50.....	\$2.14 ea.

**ATHANA:** The diskette you've used hundreds of times...without knowing it. ATHANA specializes in producing top-of-the-line diskettes for software manufacturers.

Come's disk with Typec sleeves, user ID labels, reinforced hubs and write-protect tabs. 100% LIFETIME WARRANTY.

5.25" 8DDO Bulk Pack of 50.....	\$ .97
5.25" 8DDO Bulk Pack of 50.....	\$1.01 ea.

**NASHUA:** One of the largest diskette manufacturers in the nation...and the leader in rapid disk packs for mainframes. Of course, there's a 100% LIFETIME WARRANTY on all NASHUA products. Typec sleeves, reinforced hubs, user ID labels and write-protect tabs are included.

5.25" 8DDO Bulk Pack in 50's.....	\$ .97 ea.
5.25" 8DDO Bulk Pack in 50's.....	\$1.07 ea.

**MEMOREX:** Is it real or is it Memorex? It's Memorex, packaged as shown, complete with Typec sleeves, reinforced hubs, user ID labels and write-protect tabs. And of course, a 100% LIFETIME WARRANTY.

5.25" 8DDO in boxes of 10.....	\$1.27 ea.
5.25" 8DDO in boxes of 10.....	\$1.69 ea.
5.25" 8DDO-2ETP diskettes in boxes of 10.....	\$1.59 ea.
5.25" 8DDO-HD for IBM PC/AT in boxes of 10.....	\$3.95 ea.
5.25" 8DDO-135TPH MicroPC copy in boxes of 10.....	\$2.49 ea.

**SONY:** The company that invented the 3.50" drive...and licenses it to every other manufacturer. Based in 10's with user ID labels. (There's no need for write-protect tabs or Typec sleeves.)

3.50" 8DDO for the Macintosh.....	\$2.55 ea.
3.50" 8DDO for Data General/One and HP.....	\$3.99 ea.

**RIBBONS:**

100m MXT-7010.....	\$3.95 ea.
100m MXT-7010 4500 S/E.....	\$4.95 ea.
100m Micro81.....	\$1.51 ea.
100m Micro84.....	\$3.79 ea.

Add 25¢ each ribbon for spools

**ORDERING INSTRUCTIONS:** Add \$3.00 shipping per 100 diskettes or fraction thereof. All other items, add shipping as shown. VISA, MASTERCARD AND PREPAID orders accepted. COD orders, add \$3.00 Special Handling fee. APO, FPO, HI, AK and PW orders, add 5% to cover Postal Postage shipping. No foreign orders. IL residents, add 7% sales tax. MINIMUM ORDER: \$35.00 or 30 diskettes.

DISK WORLD, Inc.  
(In Illinois 1-312-256-7140)

TOLL FREE: 1-800-621-6827  
629 Green Bay Rd. Wilmette, Illinois 60091

## HOW TO TURN YOUR PERSONAL COMPUTER INTO YOUR PERSONAL BANKER.

Here's how to make the investment in your personal computer really pay off.

With SPECTRUM™, the new electronic home banking and information system from The Chase Manhattan Bank, N.A., teamed up with your PC, SPECTRUM is your direct link to Chase. And the start of a better way to manage your money and your time.

A push of a button lets you pay bills electronically anywhere; transfer funds; keep records, and more. Even get vital financial information and trade stocks\* at discount rates.

Right at home, anytime—with complete security. All, including electronic mail, starting at just \$5 a month. And with our new subscriber rebates and free offers, now's an even better time to get control of your finances. Call today for your free demo diskette.



**CALL TOLL FREE  
1-800-522-7766**



**CHASE**

© 1985 The Chase Manhattan Corporation. \*Stocks are traded through Chase & Company's Chase & Paine.

umn, line, and page. You can call up a screen with control codes, but when you're writing, what you see is what you get: Underlined passages are underlined, bold is bold, indenting looks like it will on paper.

The spelling checker is another extraordinary feature; it's fast and effectively organized. The dictionary comes with 100,000 words (more than most paperback dictionaries), is easily expanded, stunningly quick, and simple to use. It even looks up words phonetically: If you type *kommity* it will look up *committee* for you. It checks for double words (such as "Paris in the the spring") and gives a word count as well.

If there are things you do repeatedly, you can create macros, little programs that control the word processor. Then, at the touch of a single key, you can type in a frequently used address, save the file, search for another file, etc. You can also create two screens and switch between them, and instruct the

word processor to automatically save backup copies of your current work every few minutes. And *WordPerfect* has a sophisticated, efficient footnote capability.

Sophisticated and efficient are good words to describe this program. While it is easy to use, it contains so many features and advanced commands that even the most demanding writers should find what they need. Statistical typing, math calculation, automated outline numbering, automated compilation of tables of contents—the list goes on. If you need some arcane function, you're likely to find it in *WordPerfect*. If you simply need a powerful, sensible writing machine, you owe it to yourself to consider this fine tool.

*WordPerfect*  
SSI Software  
325 North State Street  
Orem, UT 84057  
\$495

## Adventures In Narnia For Apple And 64

C. Regena

**Requirements:** Commodore 64 with a disk drive; or an Apple II-series computer with at least 48K RAM and a disk drive.

*Adventures in Narnia* is an adventure game based on the book *The Chronicles of Narnia* by C.S. Lewis. When we first opened the package, my son exclaimed, "Hey, I've read that book!" Inside is a paperback entitled *The Lion, the Witch and the Wardrobe*, a popular story in the *Chronicles* series which was the inspiration for *Adventures in Narnia*. The characters in the game are the same as those in the book. You don't have to read the book to play the game, but if you play the game and enjoy it, you'll probably want to read the book.

*Adventures in Narnia* is actually a combination adventure, arcade, and board game. It even comes with a deck of game cards and a pair of dice. High-resolution computer graphics replace the traditional board, but sometimes during the arcade action you bump into something that requires you to roll the dice or choose a card. *Adventures in Narnia* was designed to use the computer, but still resemble a board game and bring human interaction into play. As a result, the game is not always machine-controlled. Its authors point out that the computer waits while you "think, discuss, decide." You can "strategize and argue (without penalty) in the middle of the game, allowing the fun and humor of dialogue that is missing in normal videogames."

Other adventures are available in this series as well—such as *Dawn Treader*, based on the story *The Voyage of the "Dawn Treader."*

### Dodging Evil Dwarfs

The game actually is a two-part adventure with two different arcade screens. In the first adventure, you start out in the wardrobe and try to gain points and strength. In the second adventure, you use the strength to reach a character called Aslan the Lion.

You start the game by shuffling the cards and placing them near the computer. Your player is at the right side of the screen among randomly placed bushes, flowers, and beavers. Evil dwarfs dart about, and you have to avoid them. Dwarfs can also hide in the bushes, so you don't want to hit a bush. You can gather flowers to gain points, and you can meet a friendly beaver to gain strengths (indicated by hearts at the top of the screen). The evil dwarfs patrol Narnia and go around stomping on flowers and beavers to prevent your success.

The game action is quick. If you hit a dwarf (or a moving bush), you're sent to a dwarf battle. The dwarf thinks of a random number, and you must roll the dice to beat his number. If you win, you get 500 points, but if you lose, you sacrifice one strength.

From time to time, Edmund and the Witch appear on the screen. Your

job is to intercept Edmund to prevent him from reaching the Witch. The first adventure ends if the Witch captures Edmund, or if you get ten heart strengths, or if your time runs out. Then the next adventure starts.

### Inside The Ice Maze

The second adventure takes place in the Ice Maze with your character at the lower-right corner of the screen and Aslan the Lion at the upper left. Your object is to get to Aslan, but the Witch sends evil crystals through the maze to stop you. If you get hit by a crystal, you're sent back to the beginning and you lose one heart. If you lose all the hearts and get hit again, you lose the game.

The gravity chutes are another hazard. Snow is falling inside these chutes, and if you step into one, you'll tumble to the bottom and find yourself sealed inside the maze.

If you run into a door, you're side-tracked to a subadventure. Since the game action is fast, you'll probably run into some doors by accident. Once in a subadventure, you're directed to pick a card. Then you enter the card's code into the computer. If your card is Aslan the Lion, you automatically gain one heart strength. If you draw a Zap card, you're automatically sent back to the beginning of the maze and you lose one strength. Other cards—such as Fenris Ulf the Wolf, Cair Paravel the Castle, and Jadis the Witch—require you to roll the dice to determine your consequences, which can be good or bad. After the subadventure, you return to the maze for another crack at Aslan the Lion.

You lose the game if the Witch steals all your hearts with her ice crystals or if time runs out. You win the game by reaching Aslan the Lion. Your final score is determined by the running score on the screen plus 1000 points for foiling the Witch, 1000 points for each remaining heart, and the remaining time multiplied by 10.

The instruction booklet that comes with this package is very good. It presents all aspects of the game so you can identify each object and recognize whether it is good or bad. Color screen photos are accompanied with explanations for each possibility. And as you play the game, the screen instructions also are easy to understand.

If you own more than one computer, note that the Apple and 64 versions of this program are on flip sides of the same disk.

*Adventures in Narnia*  
Word, Inc.  
4800 W. Waco Drive  
Waco, TX 76796  
\$39.95

# Classified

## COMPUTE! Classified is a low-cost way to tell over 350,000 microcomputer owners about your product or service.

**Rates:** \$25 per line, minimum of four lines. Any or all of the first line set in capital letters at no charge. Add \$15 per line for boldface words, or \$50 for the entire ad set in boldface (any number of lines.)

**Terms:** Prepayment is required. Check, money order, American Express, Visa, or MasterCard is accepted.

**Form:** Ads are subject to publisher's approval and must be either typed or legibly printed. One line equals 40 letters and spaces between words. Please underline words to be set in boldface.

**General information:** Advertisers using post office box numbers in their ads must supply permanent address and telephone numbers. Orders will not be acknowledged. Ad will appear in next available issue after receipt.

**Closing:** 10th of the third month preceding cover date (e.g., June issue closes March 10th). Send order and remittance to: Harry Blair, Classified Manager, COMPUTE!, P.O. Box 5406, Greensboro, NC 27403. To place an ad by phone, call Harry Blair at (919) 275-9809.

**Notice:** COMPUTE! Publications cannot be responsible for offers or claims of advertisers, but will attempt to screen out misleading or questionable copy.

### SOFTWARE

**TIPPY LOTTO PICKER.** Go for Million Dollar jackpot! Picks LOTTO, Daily Numbers, & Win-4. All USA Lotto games are programmed in \$25. RIDGE, 170 Broadway, #201-C, NYC, NY 10038. 718-833-6335. Free Catalog.

**IBM PCjr OWNERS.** We have the software you want. We also have lots for easy 2nd drive addition. Free catalog. COWI, SOFTWARE, 1435 Burnley Sq. N., Columbus, OH 43229.

**PRINT USR - Formatted numeric output.** Use with your BASIC program. ML speed. 48K+ Apple or C64. Send \$9.95 to: MAK Ent., 1702 W. Lender, Mesa, AZ 85202.

**AMCUP, AN ORIGINAL** sailboat racing game 1 or 2 players. C-64 w/1541 + joystick. \$14.95 ch./m.e. SASE for info. Dream Dog Software, P.O. Box 1724, Eugene, OR 97440.

**"CRAPSMAN"** The C-64 casino craps tutor! Learn all bets. Test systems. Up to four players \$31.95 pp. disk. EMCEE SOFT CORP., P.O. Box 402124, Miami Beach, FL 33140.

**SHOPPING LIST PROGRAM** for C-64 with 1541 disk and printer. It prepares a shopping list for meals you select, organized by your market sections. Send \$42 to 4955 Woodman Ave., Sherman Oaks, CA 91423.

**SOFTWARE PROTECTOR 64.** Copy and list protect your valuable BASIC programs. Only \$10 (disk only). Systems Software, 44 Rte 25A #209, Smithtown, NY 11787.

**PINOCCHIO** for Commodore 64, PC and PCjr. Real, double-deck, partnership pinocchle gives you a partner and two opponents! On disk, \$25 postpaid. Jim Bernard, 8018 Sunset Path Ct., Springfield, VA 22153.

**TI CHECKWRITER** Letter code and dollar amount are all that's necessary to print a complete check. Provides yearly totals for taxes. Disk \$24.95. R.L.P. Software, 248 Purisma Rd., Woodside, CA 94062.

**SCIENCE SOFTWARE** with graphics for the Commodore 64. Astronomy programs, etc. Catalog and demo disk \$3. David Edge, 7952 W. Quatro Dr., Littleton, CO 80123.

**FIXED ASSET DEPRECIATION** on C64. Handles assets and pre-sets methods. \$39.95. For free info write: MPM Software Prods., P.O. Box 3522, Glendale, CA 91201.

**ATARI USERS 4 games** on cassette, \$25. Saucers + monsters, poker, sea finder, pinball. Write or call Keith Anderson, 8435 12th Ave., Silver Spring, MD 20903 (301) 434-0285.

### HARDWARE

**REPLACEMENT POWER SUPPLIES** FOR C-64 or VIC-20 \$29 ea plus \$3 shipping. 64 parts list \$5-refundable w/1st order. AA Computer, 2726 Park St., JAX, FL 32205.

**NUMERIC KEYPAD** with joystick CURSOR control, gold contact \$75. Centronics interface \$40. Autodisk cartridge automatically loads programs from easy-to-use menu \$40. All C-64 fully compatible. \$8.64 RF Mod. \$40. Battery Back-up with best available protection for C-64 \$50, Apple/IBM \$150 (U.S.) APPLIED COMPUTING, Box 15466, PTBO, ONT. K9J 7E7. Send \$2 for cat. Call (705) 745-8617 anytime.

### MISCELLANEOUS

**Trade in your used Commodore or Atari** on a brand new C-128 or Atari ST. This offer may not be available through retail outlets. Brochure \$1.90 and SASE. NEW WEST TECHNOLOGY, 48 Monroe Pkwy., Box 20, Ste. 134, Lake Oswego, OR 97034.

**FLYING COMPUTERS** but stuck on the ground? Learn requirements to fly aircraft, with or without license. \$3.00 to Aviation Resource, P.O. Box 369-C, Lancaster, NY 14086.

**Super 99 Monthly**—quality info for TI-99/4A & compatibles since 09/84. US \$12/yr. Other \$16/yr. Bytemaster Computer Sales, 171 Mustang, Sulphur, LA 70663.

**FREE SHOP BY MODEM** in our revolutionary electronic shopping mall. We even accept credit cards! We have products YOU want! Call 1-818-840-8066 with your modem NOW!

**BUY-SELL-TRADE** computers, hardware, software. Announce clubs, BBS's, etc. Send for free sample of Micro-Swap, the computer classifieds. P.O. Box 24, Diamond, IL 60129.

**COMPUTER OWNERS!!** Earn dollars and save money with your computer. Write C&D Associates, Box 851, Mt. Prospect, IL 60056. Specify computer type. No investment!

**TI-99/4A Software/Hardware bargains.** Hard to find items. Huge selection. Fast Service. Free Catalog. D.E.C., Box 690, Hicksville, NY 11801

**LOWEST DISK PRICES \$5/DK** with sleeve & label-10/\$8.99, bulk 100/\$79.50/DK with sleeve & label-10/\$10.90, bulk 100/\$99. Prime quality major mfr's overstock! Money-back satisfaction guarantee! Min order \$15. Pay by MC/Visa/AE, UNITECH 20 Hurley St., Cambridge, MA 02141. (800) 343-0472, in Mass (617) 864-8324.

### HELP IS ON THE WAY!

Just call 1-800-334-5848 to get your free copy of the latest COMPUTE! Books Catalog! If you need help in getting information on all of the latest COMPUTE! book titles available plus all COMPUTE! backlist titles, call us today!

**SAFEGUARD YOUR COMPUTER KEYBOARD** from contaminants with OVERBYTE molded keyboard covers. \$24.95 (apart & durable). Call (212) 866-2583/send \$2 further info. OVERBYTE-P.O. Box 10652, Burbank, CA 91510.

**PHONEMARK DATASETTES:** C-64, Plus4, Vic20. \$25.00 plus shipping. We are direct importers. American Mercantile Co., 2450 1st Ave., South, Seattle, WA 98134 (206) 824-6141.

**DISK SERVICE MANUAL.** Service floppies without special software or equipment. FREE Information. Consumertronics-DSM, P.O. Drawer 537, Alamogordo, NM 88310.

**MAXELL MDI, \$1.39 - MD2, \$1.99.** Dysan 104/110, \$1.79 - 104/22D, \$2.39. Shipping \$3.75. Also Verbatim, IBM, 3M, BASF TAPE WORLD, 220 Spring St., Butler, PA 16001, 1-800-245-6050, Visa, MC.

### \$5 MONEY MAKER \$5

Increase your income by hundreds and even thousands of dollars. Money back guarantee! (Apple, Macintosh, IBM) Free brochure: 1-800-223-5838 (305-771-6496).



# ANIMATOR

Steve Johnson

*This feature-packed utility makes it a breeze to create your own short cartoons or animation sequences on the computer screen. The original version is for the IBM PC with BASICA and color/graphics adapter, or Enhanced Model PCjr with Cartridge BASIC. We've added versions for the Atari 400/800, XL, and XE with at least 32K RAM for disk or 24K RAM for tape; Commodore 64 and 128 (in 64 mode); TI-99/4A with Extended BASIC; and Apple II-series computers with at least 48K RAM. The Atari and 64 versions also require a joystick.*

Computer animation can be marvelous to behold but a drudge to produce. Whether you're working in BASIC or machine language, creating objects and manipulating them on the screen can mean fumbling for hours with PEEKs, POKEs, bits, bytes, and other tedious details.

"Animator" goes a long way toward automating this process. It works much like a cartoonist's sketchpad, letting you draw a series of similar images which are then displayed in rapid sequence to create the illusion of movement. Your finished cartoons can be saved on disk or tape and reloaded for viewing later.

# What every Apple II+ and IIe user should ask before buying the "Sider" 10 MB hard disk:

When a company offers a superior quality 10 megabyte Winchester hard disk for only \$695, it's bound to raise a few eyebrows...and a lot of questions. The fact is, you're probably already wondering "Can I really get a 10 megabyte hard disk that's *reliable* for only \$695?" The answer is: **ABSOLUTELY**...when you choose the Sider from First Class Peripherals.

## What's so great about the Sider?

For starters, the Sider lets you boot your *Apple II+ or IIe* directly off the hard disk—unlike some other Winchester subsystems. Rebooting is also trouble-free. And the disk is partitionable, allowing you to allocate space to four operating systems on the same disk. The Sider supports: Apple DOS 3.3; Pro DOS™; Apple Pascal; and CP/M®.

What's more, a small "footprint" lets you incorporate the compact Sider into your existing computer set-up with ease.

In addition, with the Sider, you not only pay far less for the subsystem, you also save money on installation. Because, unlike other 10 MB systems that require the purchase of expensive "extras," the Sider is *plug and play*. Everything you need is provided, including cables, host adaptor, installation software and manual.

## What makes it so reliable?

To start, the Sider is manufactured, and sold exclusively, by First Class Peripherals, an innovative computer company which is backed by Xebec. The computer industry's leading manufacturer of disk controllers, Xebec has over a decade of experience serving customers like IBM, Toshiba, Texas Instruments and Hewlett Packard. It's this kind of expertise that helps assure the Sider's performance.

Special design features further enhance reliability. The Sider's controller is the field-proven, industry standard Xebec S410A. And Xebec's 3200 drive tester, the



## "Only \$695?"

toughest in the industry, ensures that the Sider will operate reliably. One more assurance of the \$695 Sider's quality: it's UL Approved and FCC Class B rated.

## But why is it only \$695?

You pay less for the Sider than for other 10 MB hard disks simply because you're paying for the superior quality components inside the unit, not for a lot of retail overhead costs. Since First Class Peripherals sells direct, you avoid dealer and dis-

tribution expenses, and pay only for the product.

## What about a guarantee?

Like many experienced Apple users, you may be reluctant to buy a hard disk priced at only \$695 without first seeing for yourself how it performs. That's why First Class Peripherals offers you a reassuring, money-back guarantee that eliminates any risk on your part. Simply order the Sider and use it for 15 days. Then, if you're not entirely satisfied, return it and receive a full refund—no questions asked.

The Sider also comes with a full one-year limited warranty. Plus, there's a convenient, toll-free hotline you can call anytime you have a technical or service question, or need help.

## Don't delay.

## Order the Sider now.

To receive the Sider 10 megabyte Winchester hard disk subsystem for only \$695, simply order using the coupon below. For faster service, order by phone and charge to your VISA, MasterCard or American Express. (You can also call us if you have any questions or technical concerns about the Sider. We'll see you get the help you need.) Call toll-free:

# 1 800 538-1307

Extension 709

☐ **Yes**, please send me the Sider, including half-height 10 megabyte Winchester hard disk drive, Apple adaptor board, cable, complete installation software and documentation.

I prefer to pay as follows:

- ☐ I've enclosed my check or money order for \$695\* + \$15 shipping and handling, payable to First Class Peripherals.
- ☐ Please bill the following credit card account for \$695\* + \$15 shipping and handling
- ☐ VISA ☐ MasterCard ☐ American Express

Card # \_\_\_\_\_ Exp. Date \_\_\_\_\_

Signature \_\_\_\_\_  
\*Residents of CA, NY and PA, please add appropriate sales tax

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_  
State \_\_\_\_\_ Zip \_\_\_\_\_  
Telephone (area code) \_\_\_\_\_  
Mail to \_\_\_\_\_

**FIRST CLASS PERIPHERALS**

3579 Highway 50 East  
Carson City, NV 89701

Apple and Pro DOS are trademarks of Apple Computer, Inc.  
CP/M is a registered trademark of Digital Research, Inc.

If you're using the IBM PC/PCjr version of Animator, type in and save Program 1, then run it and follow the instructions below. *If you have an Atari, 64, TI, or Apple, you should also read the following instructions as a general guide to using Animator.* Then refer to the typing instructions and programming notes specific to your computer for additional details.

## Drawing An Image

When you run Animator, it displays an editing screen with 20 numbered frames. You can draw as many as 20 pictures, one in each frame, then flip rapidly through the frames to create animation. The frame number displayed at the upper left of the screen shows which frame you're currently working on. Normally, Animator begins the animation with frame 1 and ends with frame 20. But you can start and end the animation wherever you like. For example, a short sequence might start with frame 1 and end with frame 3. To view only part of a long sequence, you might start at frame 12 and end at frame 18, and so on. The frame number is controlled by pressing the right and left arrow keys.

The frame number also determines which frame you'll be working on when you go to the editing screen. Let's start with a simple example. Make sure the frame number is set to 1, then press the 2 key to select the editing function and press Enter at the next prompt.

After a brief pause, Animator displays a drawing grid with a blinking cursor. Edit mode has three main functions, selected by pressing different keys. Press D to draw with the cursor, E to erase, and M to move the cursor without disturbing anything on the screen.

Draw a simple shape on the grid to become familiar with these basic functions. As you'll see, Animator displays the shape in its actual size to the left of the drawing grid. An inverse function lets you reverse everything on the grid—every dot becomes a blank, and vice versa (be patient—it takes Animator about a minute to complete this process).

Once the picture is finished, you can press S to save it and return to the main screen. *Note that you must save a picture with S to put it in the frame.* If you exit the edit mode by pressing Q, the new picture is lost and Animator uses whatever that frame previously contained. Try drawing a simple shape and saving it with S (since this is just for practice, any scribble will do). When you return to the main screen, Animator displays the picture in frame 1.

## Frame By Frame

Now you're ready to draw the next frame in the sequence. In most cases you'll want to make only slight changes from one frame to the next, to simulate smooth motion. To save time, Animator lets you copy a picture from one frame to another. Let's demonstrate this by copying the picture from frame 1 to frame 2. Set the picture number to 1 with the arrow keys, then press 2 to edit. Animator displays a prompt, inviting you to enter a frame number. To edit the current picture number, you would just press Enter. However, by entering a different number you can copy the current picture into a different frame, then change that picture to make the next frame in your cartoon.

When you enter 2 at the prompt, Animator copies the picture from frame 1 into the drawing grid. When the drawing grid appears, make some change in the picture to distinguish it from frame 1. Now press S to save the picture in frame 2 and return to the main

screen. Animator displays both pictures in their respective frames.

After drawing a few frames, you're ready to bring them to life. The first step is to specify the starting and ending frame numbers. The starting number determines which frame begins the animation, and the ending number tells Animator where the series ends.

Set the starting number first. Use the arrow keys to set the frame number to 1, then press the 3 key. Now use the arrow keys to make the frame number match the last frame that contains a picture, then press the 4 key. This sets the ending number. You must always set the starting and ending numbers before selecting animation (if you don't, Animator flips through all 20 frames whether they contain pictures or not). Once these numbers are set, press the 1 key to view the sequence. Press the space bar to pause and Enter to stop it.

By selecting different speed and pause values, you can move the animated figure across the screen. The speed value can range from -15 to 15. When it is 0, the figure is animated in place; positive values move the figure from left to right, and negative values move it from right to left. The greater the value, the faster the figure moves. Press the 5 key to decrease the animation speed, and 6 to increase it.

The pause value controls the time delay between each frame of the animation. A small pause value makes the pictures change very quickly, while larger values slow down the process.

## Macro Editing Features

Animator provides a few macro (large-scale) editing features to help you work with longer cartoons. The insert function lets you insert a blank frame anywhere in the series. To use it, set the frame number to the number of the frame where you want to insert a blank, then press the I key. The designated picture (and all those following it) are bumped forward one frame. Note that the picture in frame 20 is always lost when you insert.

The delete function lets you delete any frame in the series. Change the picture number to the frame you want to eliminate, then press D. All the higher numbered

## FREE CATALOG!

Features Precision Tools and Equipment for Computers



Jensen's new catalog features hard-to-find precision tools, tool kits, tool cases, test equipment and computer accessories used by sophisticated hobbyists, scientists, engineers, laboratories and government agencies. Call or write for your free copy today.

**JENSEN  
TOOLS INC.**

7815 S. 46th Street  
Phoenix, AZ 85040  
(602) 968-6231



**88K** *Lowest Price In The USA!* **152K**

# Computer System Sale

• Students • Word Processing • Home • Business

**88K System \$449**  
**152K System \$499\***



EDUCATE WITH ATARI



<sup>1</sup> **Rated "Best Buy" by Consumers Digest Buyers Guide, January 1985**

LOOK AT ALL YOU GET FOR ONLY

**\$449**  
SYSTEM PRICE

LIST PRICE

INDIVIDUAL  
SALE PRICE

Atari 800XL 88K Computer  
 Atari 1050 127K Disk Drive  
 Atari 1027 Letter Quality 20 CPS Printer  
 Atari Writer Word Processor  
 Atari BASIC Tutorial Manual

\$179.00	\$109 <sup>00</sup>
299.00	199 <sup>00</sup>
299.00	199 <sup>00</sup>
59.95	39 <sup>95</sup>
16.95	12 <sup>95</sup>

SAVE \$100  
 ALL 3 ONLY  
**\$449<sup>00</sup>**  
 SYSTEM  
 SALE PRICE

**152K SYSTEM**  
**\$499<sup>00</sup>**

All connecting cables & T.V. interface included  
 Monitors sold separately

**TOTALS**

**\$852.90 \$559.90**

## Other Accessories

12" Hi Resolution Green or Amber Screen Monitor  
 13" Hi Resolution Color Monitor  
 \* ATARI 130XE 152K Computer

List	Sale
\$199.00	<b>69.95</b>
\$399.00	<b>169.95</b>
\$249.00	<b>159.00</b>

Add \$9.95 for  
 Connection Cables  
 (Monitors Only)  
 (Ltd. Qty)  
 Add \$10 for UPS

**15 DAY FREE TRIAL** We give you 15 days to try out this ATARI COMPUTER SYSTEM<sup>1</sup>! If it doesn't meet your expectations, just send it back to us prepaid and we will refund your purchase price! **90 DAY IMMEDIATE REPLACEMENT WARRANTY** If any of the ATARI COMPUTER SYSTEM equipment or programs fail due to faulty workmanship or material within 90 days of purchase we will replace it IMMEDIATELY with no service charge!<sup>1</sup>

**Best Prices • Over 1000 Programs and 500 Accessories Available • Best Service**  
**• One Day Express Mail • Programming Knowledge • Technical Support**

**Add \$25.00 for shipping and handling!!**

Enclose Cashiers Check, Money Order or Personal Check. Allow 14 days for delivery. 2 to 7 days for phone orders. 1 day express mail!  
 We accept Visa and MasterCard. We ship C.O.D. to continental U.S. addresses only. Add \$10 more if C.O.D.

**COMPUTER DIRECT**

*We Love Our Customers*

22292 N. Pepper Rd., Barrington, Ill. 60010

**312/382-5050 to order**

pictures move down one frame, deleting the picture in the designated frame. Frame 20 is always blank after a deletion.

The inverse function (press 9) works just like inverse in editing mode, but inverts all 20 frames at once.

To clear all 20 frames, press Q to quit or C to clear. Since these last two functions can have drastic results, Animator lets you abort either one without harm.

When you finish a sequence, press S to save it on disk. The screen clears and displays three options: You can Press A to abort the save, F to list the picture files on that disk, or any other key to continue with the save. Picture filenames are limited to eight characters (the first character cannot be a number). Do not add a three-character extension; Animator automatically appends the extension .ANI when you save or load a picture file.

Finally, Animator's program option (available only in the IBM PC/PCjr version) can write a separate BASIC program to display your cartoon. Press P to select this option, and sit back while Animator writes the new program to disk under the filename PRG.BAS. Afterward, Animator ends with a reminder to reload PRG.BAS and save it with a new filename. This prevents the program from being overwritten if you select this option again.

## Commodore 64 Version

The 64 version of Animator is written entirely in machine language, but you can use it without understanding machine language at all. Type in and save Program 2 using the "MLX" machine language entry program printed elsewhere in this issue. Here is the information you'll need:

Starting address: 49152  
Ending address: 52991

After you've saved "64 Animator," plug a joystick into port 2 and load the program with LOAD "filename",8,1 for disk or LOAD "filename",1,1 for tape. Type SYS 49152 and press RETURN to run the program.

64 Animator's main screen consists of 21 picture frames instead of 20 as found in the IBM

version. It also uses sprites to animate the frames. Although the Commodore 64 normally is limited to displaying eight sprites at once, 64 Animator employs as many as 22 simultaneous sprites with the raster interrupt technique described in *COMPUTE!'s First Book of Commodore 64*.

When you begin the program, the frames may contain random data; Press C to clear them out. Most program functions are controlled with the joystick. Near the bottom of the screen you'll see a list of several functions (frame numbers, options, and so on, as described above in the general instructions). As you move the joystick up or down, the function you select is highlighted in reverse video. To increase or decrease the selected value, move the joystick left or right.

Press the E key to enter edit mode. The joystick moves the blinking cursor around the drawing grid, and the fire button toggles the space under the cursor on or off. To draw or erase more than one space at a time, hold the joystick button down and move the stick in the direction you want.

The current picture number is displayed to the right of the screen. You can move to a different picture within edit mode: Press the + key to increase the picture number, and the - key to decrease it. Animator always displays the current picture in actual size above the picture number. Above and to the right of the current picture is the next picture in the series, and above to the left is the previous picture. (If you haven't drawn any pictures yet, these frames may be blank or contain random shapes.)

Press I within edit mode to invert the shape (change dots to blanks, and vice versa). The cursor keys shift the entire shape one space inside the grid, either left, right, up, or down. You can also expand the picture horizontally (press X) or vertically (press Y). Expansion is toggled off by pressing the same keys, and can be used on the main screen as well.

Edit mode lets you copy a picture from one frame into another. Press the I7 function key to store the current shape in the picture buffer. Then change the picture

number with + or - and press f8 (SHIFT-f7) to copy the picture from the buffer into the new frame. In this way you can quickly draw a series of shapes without leaving edit mode. The R key returns you from edit mode to the main screen.

## Animating On The 64

Before watching the animation, set the starting and ending frame numbers to the appropriate range. The joystick button turns animation on and off. To change the speed during animation, move the joystick left or right. Press the f1 function key to clear everything but your animated figure off the screen (f1 also brings back the main screen). You can change the picture color by pressing the CTRL key and one of the number keys from 1-8. The colors are the same as those printed on the front of the keys. For example, press CTRL-0 to color every figure black.

Insertions and deletions work as in the IBM version, except that an insertion retains the original picture in the selected frame rather than inserting a blank. To quickly fill several frames with the same picture, press I several times.

64 Animator can save and load picture files with either disk or tape. Be sure to set the starting and ending numbers before saving, since nothing is saved outside that range. The prefix ANI is added to picture files on disk; do not type this prefix when loading or saving. To abort a save or load, press RETURN.

It's possible to merge pictures from different files if the two files use different frame numbers. Simply load the second file after the first. The main screen now contains pictures from both files. If the files have overlapping frame numbers, the second file has priority. For example, say that you load ANI.A, which uses frames 1-3, then load ANI.B, which uses frames 3-5. Frame 3 will contain the picture from the ANI.B file.

64 Animator also lets you add sound effects in edit mode. Press D to add or clear the drum sound. Animator displays a small drum on the screen when the sound is present. To add musical notes in edit mode, press one of the number keys from 1-8 for low notes, and SHIFT plus 1-8 for high notes. A





# Commodore 64 Accessories



## SUPER



# Summer Sizzler Sale

Sale Positively Ends 9-10-85

Prices will go back to regular sale prices

### C128 Commodore Computer

Expandable to 512K, runs C-64, CPM, and 7.0 Programs.  
(Add \$10 Shipping)

List	Reg. Sale Price	Summer Sizzler Sale
\$349 <sup>00</sup>	\$299 <sup>00</sup>	\$289 <sup>95</sup> *

### 15 1/2" Commodore 150-170 CPS Printer

Near Letter Quality, Multiple Pin Tractor / Friction Feed  
Best Printer Value in U.S.A. (Add \$17.50 Shipping)

\$895 <sup>00</sup>	\$299 <sup>00</sup>	\$249 <sup>95</sup> *
---------------------	---------------------	-----------------------

### Commodore-64 IEEE Interface

Allows you to run Pet Peripherals on the C-64, including the  
One Megabyte Disk Drive and 15 1/2" Printer

\$109 <sup>95</sup>	\$69 <sup>00</sup>	\$65 <sup>95</sup>
---------------------	--------------------	--------------------

### Juki Printer/Typewriter

Letter Quality, daisy wheel, use as typewriter and/or printer  
(auto correction) (Add \$10 Shipping)

\$349 <sup>00</sup>	\$249 <sup>00</sup>	\$229 <sup>95</sup> *
---------------------	---------------------	-----------------------

### SCM 80 CPS Printer

Tractor / Friction 10" Famous Name Printer does Graphics  
w/Interface. (Add \$10 Shipping)

\$299 <sup>00</sup>	\$159 <sup>00</sup>	\$149 <sup>95</sup> *
---------------------	---------------------	-----------------------

### Cardco G Plus Interface

Converts Commodore to centronics for use with most printers, plus  
does Commodore graphics on graphic printers

\$109 <sup>00</sup>	\$59 <sup>00</sup>	\$49 <sup>95</sup>
---------------------	--------------------	--------------------

### Alphacom 40 Column Printer

Thermal technology - does graphics. (Add \$10 Shipping)  
Alphacom C-64 or Atari Interface \$8.95.

\$99 <sup>00</sup>	\$24 <sup>95</sup>	\$22 <sup>95</sup>
--------------------	--------------------	--------------------

### 190K Slim-Line Disk Drive

Cooler, 20% faster, quieter than 1541 drive (Add \$10 Shipping)

\$249 <sup>00</sup>	\$169 <sup>00</sup>	\$139 <sup>95</sup> *
---------------------	---------------------	-----------------------

### One Megabyte Disk Drive (1000K)

Double sided drive hooks up to C-64 with IEEE interface, perfect as a  
second drive. (Add \$10 Shipping)

\$889 <sup>00</sup>	\$199 <sup>00</sup>	\$179 <sup>95</sup> *
---------------------	---------------------	-----------------------

### 13" Premium Quality Color Monitor

Better than 1702, with separated video inputs. (Add \$14.50 Shipping)

\$399 <sup>00</sup>	\$199 <sup>00</sup>	\$169 <sup>95</sup> *
---------------------	---------------------	-----------------------

### Voice Synthesizer

Allows you to talk through your computer. Optional software lets  
you play talking adventure games (Zork, etc.)

\$89 <sup>00</sup>	\$49 <sup>00</sup>	\$39 <sup>95</sup>
--------------------	--------------------	--------------------

### 80 Column Board - 4 Slot Expander

Allows you to program in 80 columns, plus gives you a 4 slot switch  
selectable expander

\$149 <sup>00</sup>	\$79 <sup>00</sup>	\$59 <sup>95</sup>
---------------------	--------------------	--------------------

### Auto Dial Modem with Super Smart Software

Upload, Download, Print, Catalog, This Package has it all!!!  
Best Modem Value in the U.S.A.!!!

\$129 <sup>95</sup>	\$59 <sup>00</sup>	\$39 <sup>95</sup>
---------------------	--------------------	--------------------

### Musical I Software

The Best Musical Software for the C-64, allows you to change all  
parameters.

\$59 <sup>95</sup>	\$29 <sup>95</sup>	\$14 <sup>95</sup>
--------------------	--------------------	--------------------

### Oil Barrons Software

Better than Monopoly, comes with game board, disks and  
instruction manual. Strike Oil or Live in the Poor Farm

\$49 <sup>95</sup>	\$19 <sup>95</sup>	\$9 <sup>95</sup>
--------------------	--------------------	-------------------

\* plus Software Coupon Discounts

Add \$3.00 for shipping, handling and insurance. Illinois residents  
please add 6% tax. Add \$6.00 for CANADA, PUERTO RICO, HAWAII,  
ALASKA, APO-FPO orders. Canadian orders must be in U.S. dollars.  
WE DO NOT EXPORT TO OTHER COUNTRIES EXCEPT CANADA.

Enclose Cashiers Check, Money Order or Personal Check. Allow 14  
days for delivery, 2 to 7 days for phone orders, 1 day express mail!

VISA — MASTER CARD — C.O.D.

No C.O.D. to Canada, APO-FPO

## PROTECTO

We Love Our Customers

22292 N. Pepper Rd., Barrington, Illinois 60010

312/382-5244 to order

small note is displayed when a note is present. Clear a note by pressing 9. Control the duration of sounds from the main screen.

Press Q to exit 64 Animator. If you want to restart the program, you must load it again as described above. Don't reenter with SYS 49152.

## Atari Version

"Atari Animator" is in two parts. First type in and save Program 3, then Program 4. If you're using cassette, substitute the following line for line 2 of program 3:

```
##2 PRINT "COWN>PRESS RETU
RN"&:RUN "C:PART2"
```

Cassette users must also save Program 4 on the same tape immediately after Program 3. After both programs have been saved, plug in a joystick and read the general instructions above. When you're ready to continue, load and run Program 3 (be sure to rewind the tape to the beginning if you're using cassette). Program 3 loads the machine language portion of Animator, then automatically loads and runs Program 4.

The main screen provides 21 picture frames. Move the joystick up or down to select any of the values displayed at the bottom of the screen, and move it right or left to increase or decrease the selected value. Press E to go to the editing screen. In edit mode, move the blinking cursor around the drawing grid with the joystick. The fire button acts like a toggle: If you press it while the cursor is on a blank square, the cursor begins drawing; if you press it while the cursor is on a filled square, the cursor begins erasing.

To the right of the drawing grid is the current frame number and the picture in actual size. You can change to a different frame while in edit mode: Press the + key to increase the frame number and the - key to decrease it. Atari Animator also lets you copy a picture from one frame to another via the picture buffer. Press the G key to get (copy) the current picture into the buffer. Then change the frame number with the + or - key and press P to put (move) the picture from the buffer into the new frame.

Press the E key to exit the edit

mode and return to the main screen. After setting the starting and ending frame numbers, press the fire button to begin the animation. To stop the animation, press the fire button again. While the cartoon is moving, you can change the figure's color by pressing any key (shifted keys provide additional colors).

The clear function (press C) clears the current frame in edit mode or all frames on the main screen. Atari Animator does not have automatic insert or delete functions, but you can accomplish the same thing with a series of individual get and put operations. You can save sequences on disk or tape, using any appropriate filename for disk.

## Apple Version

"Apple Animator" runs on any Apple II-series computer with DOS 3.3 or ProDOS. The program requires two files on disk: the main BASIC program and a binary file (ANIMATOR2) that contains graphics data. Type in and save Programs 5 and 6, then run Program 5 to create ANIMATOR2 on disk. You must run Program 5 before running Program 6 for the first time. However, you don't need to run Program 5 each time you want use Program 6—only once to create the ANIMATOR2 file.

After running Program 5, read the general instructions above, then load and run Program 6. The program works much like the IBM PC/PCjr version, and most of its functions are self-prompting. Use the right and left arrow keys to pick the correct frame number before editing. For instance, if the frame number is 3 when you choose the edit function, frame 3 appears on the editing screen. To copy the picture in frame 3 into a different frame, enter the desired frame number when prompted.

When the editing screen appears, move the blinking cursor left, right, up, or down by pressing the J, L, I, and K keys, respectively. To put your drawing in the current frame, exit the edit mode with the save option (the quit option restores whatever that frame previously held). The remaining functions (load, save, insert, delete, invert, etc.) work as described in the gener-

al instructions above, except that Apple Animator uses no filename extenders for picture files.

## TI Version

"TI Animator" is very similar to the IBM version. Be sure TI Extended BASIC is plugged in before typing and saving Program 7.

Since the TI-99/4A screen can display only 10 frames at once, the 20 frames are divided into two groups (1-10 and 11-20) on alternate screens. Press the N key to switch back and forth. You can change the color of the screen background or foreground drawing color by pressing the B or F keys.

TI Animator can save animation files on disk or tape. When saving on tape, enter CS1 for the filename. When saving or loading from disk, be sure to type the prefix DSK1 at the beginning of each filename.

## Program 1: Animator for IBM PC/PCjr

Please refer to "COMPUTER'S Guide to Typing in Programs" before entering the listing.

```

OF 5 DEF SEG=>POKE 1047,PEEK(10
47) OR 64
K 30 KEY OFF:CLS:SCREEN 1:DEF S
=>POKE 4M4E,1
Q 110 DEF SEG=>POKE 4M4E,2
Q 130 REM *** MAIN PROGRAM ***
$
# 160 CLS:SCREEN 2:KEY OFF:FOR
I=1 TO 10:KEY I,"":NEXT
J 170 REM *** SET UP VARIABLES
***
@ 180 DIM AX(144),BX(144),CX(14
4),EX(144),FX(144),GX(144
),HX(144),IX(144),JX(144),
KX(144),LX(144),MX(144),
NX(144),OX(144),PX(144),
QX(144),RX(144),SX(144),
TX(144),UX(144),VX(144),W(2
0,54):NUM=1:STA=1:EN=20:S
P=>PL=1:GET(1,10):PRINT 1,UX
$ 190 X=Y=31:LOCATE 1,39:PRIN
T"SPED="&SP:LOCATE 1,1
:PRINT"NUMBER="&NUM:LOCA
TE 1,14:PRINT"START="&STA
:LOCATE 1,26:PRINT"STOP="
&EN:LOCATE 1,55:PRINT"C
HANGE PAUSE="&FAST
# 200 REM *** SET UP SCREEN $
$
# 210 LOCATE 22,1:PRINT"1-ANIM
ATE 2-EDIT 3-STA
RT PIC. 4-END
PIC."&LOCATE 23,1:PRINT"5
-SLOWER 6-FASTER
7-FASTER PIC. SWITCH 8
-SLOWER PIC. SWITCH":LOC
ATE 24,1:PRINT"9-INVERSE
ALL THE PICTURES"
# 220 LOCATE 21,1:PRINT"(S)AVE
(L)OAD (D)LEA
R ALL (D)UIT
(P)ROGRAM":LOCATE 4,
79:FOR I=1 TO 10:PRINT"
```

```

      "I);NEXT;LOCATE 13,79;
      FOR I=1 TO 29:PRINT "
      I);NEXT
# 230 LOCATE 29,1:PRINT"(I)NBER
      T PIC. (O)ELE
      T PIC."
# 240 FOR T=1 TO 2:FOR I=1 TO 5
      :LINE(A,X)-(A+55,Y),,0:LI
      NE(A+60,X)-(A+115,Y),,0:A
      =A+120:NEXT;A=0:I=X-80:Y=10
      :NEXT
# 250 REM *** READ PICTURES ***
      **
# 260 GET(1,10)-(54,30),AX:GET(
      61,10)-(114,30),BX:GET(12
      1,10)-(174,30),CX:GET(181
      ,10)-(234,30),DX:GET(241,
      10)-(294,30),EX:GET(301,1
      0)-(354,30),FX:GET(361,10
      )-(414,30),GX:GET(421,10)
      -(474,30),HX:GET(481,10)-
      (534,30),IX:GET(541,10)-(
      594,30),JX
# 270 GET(1,81)-(54,101),KX:GET
      (61,81)-(114,101),LX:GET(
      121,81)-(174,101),MX:GET(
      181,81)-(234,101),NX:GET(
      241,81)-(294,101),OX:GET(
      301,81)-(354,101),PX:GET(
      361,81)-(414,101),QX:GET(
      421,81)-(474,101),RX:GET(
      481,81)-(534,101),SX:GET(
      541,81)-(594,101),TX
# 280 REM *** WAIT FOR KEY ***
      *
# 290 AS=INKEY$:IF AS="" THEN 2
      90 ELSE IF LEN(AS)=2 THEN
      800 ELSE IF VAL(AS)>0 AND
      0 VAL(AS)<18 THEN 430
# 300 IF AS="P" THEN 300
      IF AS="Q" THEN 300
# 310 LOCATE 18,1:PRINT"ARE YOU
      SURE YOU WANT TO QUIT?"
      (Y/N)
# 320 AS=INKEY$:IF AS="" THEN 3
      30 ELSE IF AS="Y" THEN SC
      REEN 0,0:CLS:END ELSE LC
      LOCATE 18,1:PRINT STRING$(
      50,32):GOTO 290
# 330 REM *** CLEAR SCREEN ***
      *
# 340 IF AS<"C" THEN 300
      LOCATE 17,1:PRINT"ARE YOU
      SURE (Y/N):DEF BGS:POKE
      100,0
# 350 AS=INKEY$:IF AS="" THEN 3
      70 ELSE IF AS="Y" THEN CL
      0:GOTO 190 ELSE LOCATE 17
      ,1:PRINT STRING$(66,32):G
      OTO 290
# 360 IF AS="L" THEN 870 ELSE I
      F AS="M" THEN 990
# 370 IF AS="O" THEN 2000 ELSE
      IF AS="I" THEN 2360
# 400 GOTO 290
# 410 IF FAST=0 THEN BEEP:GOTO
      290 ELSE FAST=FAST+1:LOCA
      TE 1,60:PRINT FAST:GOTO
      290
# 420 IF FAST=150 THEN BEEP:GO
      TO 290 ELSE FAST=FAST+1:L
      OCATE 1,60:PRINT FAST:GO
      TO 290
# 430 ON VAL(AS) GOTO 520,1000,
      470,500,400,450,410,420,3
      00
# 440 REM *** SET SPEED ***
# 450 IF SP=15 THEN BEEP ELSE S
      P=SP+1:LOCATE 1,46:PRINT
      SP
# 460 GOTO 290
# 470 STA=NUM:LOCATE 1,20:PRINT
      STA:GOTO 290
# 480 IF SP=-15 THEN BEEP ELSE

```



"Animator for IBM PC/PCjr" takes advantage of extended BASIC graphics features such as GET and PUT.

```

      SP=SP-1:LOCATE 1,46:PRINT
      SP
# 490 GOTO 290
# 500 EN=NUM:LOCATE 1,31:PRINT
      EN:GOTO 290
# 510 REM *** ANIMATE PICTURES ***
# 520 LOCATE 17,1:PRINT"PRESS S
      PACE BAR TO PAUSE MOVEMEN
      T":LOCATE 10,1:PRINT"USE
      ARROW KEYS FOR SPEED":IF
      STAX THEN GO=-1 ELSE GO
      =1
# 530 FOR I=STA TO EN STEP GO
      :
# 540 PL=PL+1 IF PL=SP<1 THEN PL
      =570 ELSE IF PL+SP>500 TH
      EN PL=1
# 550 PL=PL+SP
# 560 LOCATE 1,8:PRINT I:ON I G
      OTO 630,640,650,660,670,6
      80,690,700,710,720,730,74
      0,750,760,770,780,790,800
      ,810,820
# 570 FOR N=0 TO FAST*4:NEXT:A=
      =INKEY$:IF AS="" THEN 23
      60 ELSE IF AS<" " AND LEN
      (AS)<2 THEN LOCATE 1,8:PR
      INT NUM:LOCATE 17,1:PRIN
      T STRING$(150,32):GOTO 29
      0
# 580 IF LEN(AS)<2 THEN 620
      C=ASC(RIGHT$(AS,1)):IF C=
      77 THEN SP=SP+1 ELSE IF C
      =75 THEN SP=SP-1
# 590 IF SP=-16 THEN SP=-15 EL
      SE IF SP=16 THEN SP=15
# 610 LOCATE 1,46:PRINT SP
# 620 NEXT:GOTO 530
# 630 PUT(PL,52),UX,PSET:PUT(P
      L,52),AX:GOTO 570
# 640 PUT(PL,52),UX,PSET:PUT(P
      L,52),BX:GOTO 570
# 650 PUT(PL,52),UX,PSET:PUT(P
      L,52),CX:GOTO 570
# 660 PUT(PL,52),UX,PSET:PUT(P
      L,52),DX:GOTO 570
# 670 PUT(PL,52),UX,PSET:PUT(P
      L,52),EX:GOTO 570
# 680 PUT(PL,52),UX,PSET:PUT(P
      L,52),FX:GOTO 570
# 690 PUT(PL,52),UX,PSET:PUT(P
      L,52),GX:GOTO 570
# 700 PUT(PL,52),UX,PSET:PUT(P
      L,52),HX:GOTO 570
# 710 PUT(PL,52),UX,PSET:PUT(P
      L,52),IX:GOTO 570
# 720 PUT(PL,52),UX,PSET:PUT(P
      L,52),JX:GOTO 570
# 730 PUT(PL,52),UX,PSET:PUT(P
      L,52),KX:GOTO 570
# 740 PUT(PL,52),UX,PSET:PUT(P
      L,52),LX:GOTO 570
# 750 PUT(PL,52),UX,PSET:PUT(P
      L,52),MX:GOTO 570
# 760 PUT(PL,52),UX,PSET:PUT(P

```

```

      L,52),NX:GOTO 570
# 770 PUT(PL,52),UX,PSET:PUT(P
      L,52),OX:GOTO 570
# 780 PUT(PL,52),UX,PSET:PUT(P
      L,52),PX:GOTO 570
# 790 PUT(PL,52),UX,PSET:PUT(P
      L,52),QX:GOTO 570
# 800 PUT(PL,52),UX,PSET:PUT(P
      L,52),RX:GOTO 570
# 810 PUT(PL,52),UX,PSET:PUT(P
      L,52),SX:GOTO 570
# 820 PUT(PL,52),UX,PSET:PUT(P
      L,52),TX:GOTO 570
# 830 C=ASC(RIGHT$(AS,1)):IF C=
      77 THEN NUM=NUM+1 ELSE IF
      C=75 THEN NUM=NUM-1
# 840 IF NUM=0 THEN NUM=20 ELSE
      IF NUM=21 THEN NUM=1
# 850 LOCATE 1,8:PRINT NUM:GOTO
      290
# 860 REM *** SAVE PICTURE ***
      *
# 870 CLS:ON ERROR GOTO 2500
# 880 LOCATE 10,10:PRINT"F-FILE
      S A-BORT SAVE AN
      Y OTHER KEY TO CONTINUE"
# 890 AS=INKEY$:IF AS="" THEN 0
      90 ELSE IF AS="F" THEN FI
      LES".ANI" ELSE IF AS="A"
      THEN GOSUB 2310:GOTO 190
# 900 PRINT:PRINT:PRINT"INPUT"
      NAME OF FILE TO SAVE":AS
      :IF AS="" THEN GOSUB 2310:
      GOTO 190
# 910 IF INSTR(AS,".")<0 THEN
      CLS:LOCATE 9,10:PRINT"NO
      EXTENSION PLEASE..":GOTO
      880
# 920 IF LEN(AS)>0 THEN CLS:LOC
      AT 9,10:PRINT"NO MORE TH
      AN 8 CHARACTERS PLEASE":G
      OTO 880
# 930 IF VAL(RIGHT$(AS,1))>0 OR
      RIGHT$(AS,1)="" THEN CL
      S:LOCATE 9,10:PRINT"THE F
      IRST CHARACTER CAN'T BE A
      NUMBER..":GOTO 880
# 940 GOSUB 2310
# 950 AS=INKEY$.ANI"
# 960 DEF BGS:POKE100,0:SAVE AS,0
      ,0:PRINT"IT HAS BEEN
      SAVED. PRESS ANY KEY
      TO CONTINUE":PRINT:PRINT
      :PRINT
# 970 AS=INKEY$:IF AS="" THEN 9
      80 ELSE CLS:GOSUB 2310:GO
      TO 190
# 980 REM *** LOAD PICTURE ***
      *
# 990 CLS:ON ERROR GOTO 2500
# 1000 LOCATE 10,10:PRINT"F-FILE
      S A-BORT LOAD
      ANY OTHER KEY TO CONTINU
      E"
# 1010 AS=INKEY$:IF AS="" THEN
      1010 ELSE IF AS="F" THEN
      FILES".ANI" ELSE IF AS
      ="A" THEN GOSUB 2310:GOT
      O 190
# 1020 PRINT:PRINT:PRINT"INPUT"
      NAME OF FILE TO LOAD":AS
      :IF AS="" THEN GOSUB 231
      0:GOTO 190
# 1030 IF INSTR(AS,".")<0 THEN
      CLS:LOCATE 9,10:PRINT"NO
      EXTENSION PLEASE..":GO
      TO 1000
# 1040 IF LEN(AS)>0 THEN CLS:LO
      CAT 9,10:PRINT"NO MORE
      THAN 8 CHARACTERS PLEASE
      ":GOTO 1000
# 1050 IF VAL(RIGHT$(AS,1))>0 O
      R RIGHT$(AS,1)="" THEN
      CLS:LOCATE 9,10:PRINT"TH

```

```

E FIRST CHARACTER CAN'T
BE A NUMBER..:GOTO 1000
K 1000 A=AS+";ANI":DEF SEG=BHB
BHB=BLD+AA,B,DIN ERROR
GOTO 0:GOTO 190
L 1070 REM *** EDIT A PICTURE
***
J 1000 LOCATE 16,1:PRINT"TYPE I
N 21 TO ABORT":PRINT"NUM
BER BET AT THE TOP OF TH
E SCREEN IS PIC. TO READ
FROM RETURN FOR SAME":L
OCATE 18,1:INPUT"EDIT PI
CTURE NUMBER":B:IF B<0 F
R B>21 THEN BEEP:GOTO 10
00
K 1090 IF B=21 THEN CLS:BOSUB 2
310:GOTO 190
J 1100 IF B=0 THEN B=NUM
K 1110 REM *** PUT PICTURE TO
EDIT IN SCREEN ***
P 1120 CLS:LOCATE 1,24:PRINT"MA
IT....":DN NUM GOTO 1130,
1140,1150,1160,1170,1180,
1190,1200,1210,1220,123
0,1240,1250,1260,1270,12
80,1290,1300,1310,1320
K 1130 PUT(1,50),A:GOTO 1340
N 1140 PUT(1,50),B:GOTO 1340
M 1150 PUT(1,50),C:GOTO 1340
P 1160 PUT(1,50),D:GOTO 1340
Q 1170 PUT(1,50),E:GOTO 1340
R 1180 PUT(1,50),F:GOTO 1340
S 1190 PUT(1,50),G:GOTO 1340
T 1200 PUT(1,50),H:GOTO 1340
U 1210 PUT(1,50),I:GOTO 1340
V 1220 PUT(1,50),J:GOTO 1340
W 1230 PUT(1,50),K:GOTO 1340
X 1240 PUT(1,50),L:GOTO 1340
Y 1250 PUT(1,50),M:GOTO 1340
Z 1260 PUT(1,50),N:GOTO 1340
A 1270 PUT(1,50),O:GOTO 1340
B 1280 PUT(1,50),P:GOTO 1340
C 1290 PUT(1,50),Q:GOTO 1340
D 1300 PUT(1,50),R:GOTO 1340
E 1310 PUT(1,50),S:GOTO 1340
F 1320 PUT(1,50),T:GOTO 1340
P 1330 REM *** GET ON-OFF POIN
TS ***
J 1340 FOR I=1 TO 26:FOR X=1 TO
54:A(I,X)=POINT(X,1+49)
A 1350 NEXT I
J 1360 REM *** DRAW EDITING SC
REEN ***
J 1370 FOR I=1 TO 28:LOCATE 3+I
,14:PRINT "....."
.....":IFDR
J=1 TO 54:IF A(I,J)=1 TH
EN LOCATE 3+I,13+J:PRINT
"0"
P 1380 NEXT I
Q 1390 LOCATE 1,30:PRINT"(O)UIT
(O)RAW (M)OVE (E)
RASE (C)LEAR (S)AVE
(I)NVERSE"
A 1400 GOTO 1500
P 1410 REM *** PLACE CURSOR &
**
K 1420 BLINKX=(BLINKX+1) MOD 20
:IF BLINKX<10 THEN 1470
ELSE 1440
P 1430 REM *** CURSOR OFF ***
K 1440 IF A(ROW,COLUMN)=0 THEN
CHS="." ELSE IF A(ROW,C
OLUMN)=1 THEN CHS="0"
P 1450 GOTO 1480
K 1460 REM *** CURSOR ON ***
L 1470 IF CURS=1 THEN CHS="."
ELSE IF CURS=0 THEN CHS
="0" ELSE IF CURS=1 THEN
CHS="+"
E 1480 LOCATE 3+ROW,13+COLUMN:P
RINT CHS:RETURN

```

```

K 1490 REM *** REMOVE CURSOR
***
N 1500 IF A(ROW,COLUMN)=0 THEN
CHS="." ELSE IF A(ROW,C
OLUMN)=1 THEN CHS="0"
O 1510 LOCATE 3+ROW,13+COLUMN:P
RINT CHS:RETURN
P 1520 LOCATE 24,18:PRINT "wait
":FOR I=1 TO 28:LOCATE
3+I,14:PRINT STRING$(54,
46):
E 1530 NEXT I:ERASE A:OM A(20,54
):LOCATE 24,18:PRINT"
":PUT(1,50),UX,PSET:RE
TURN
O 1540 REM *** GET CURSOR ***
J 1550 ROW=1:COLUMN=1:CURS=0
N 1560 REM *** MAIN LOOP ***
E 1570 BLINKX=0:IF CURS=1 THEN
A(ROW,COLUMN)=PSET(CO
LUMN,ROW+49),0 ELSE IF C
URS=+1 THEN A(ROW,COLUM
N)=1:PSET(COLUMN,ROW+49),
1
L 1580 BOSUB 1420
P 1590 A=INKEY$:DEF SEG=POKE 1
00,0:IF LEN(A$)=0 THEN 1
580 ELSE IF LEN(A$)=1 TH
EN 1600 ELSE IF LEN(A$)=
2 THEN 1720
O 1600 CODE1=ASC(A$) AND 15HF
P 1610 REM *** READ KEYS ***
F 1620 IF CODE1=ASC("E") THEN 2
040
P 1630 IF CODE1=ASC("M") THEN 2
050
F 1640 IF CODE1=ASC("D") THEN 2
060
K 1650 IF CODE1=ASC("C") THEN 2
080
F 1660 IF CODE1=ASC("S") THEN 2
100
E 1670 IF CODE1=ASC("O") THEN 6
DSUB 2310:GOTO 190
P 1680 IF CODE1=ASC("I") THEN 1
710
O 1690 GOTO 1500
E 1700 REM *** INVERSE A PICTU
RE ***
F 1710 GET(1,50)=(-54,70),UX:PUT
(1,50),UX,PSET:GET(1,7
5)=(-54,95),UX:GOTO 1340
E 1720 IF ASC(A$)>0 THEN 1570
ELSE CODE2=ASC(RIGHT$(A$
,1)):BOSUB 1500
N 1730 REM *** READ ARROW KEYS
***
K 1740 IF CODE2=71 THEN 1840
M 1750 IF CODE2=73 THEN 1870
M 1760 IF CODE2=79 THEN 1900
U 1770 IF CODE2=81 THEN 1930
P 1780 IF CODE2=72 THEN 1960
F 1790 IF CODE2=75 THEN 1980
K 1800 IF CODE2=77 THEN 2000
O 1810 IF CODE2=80 THEN 2020
O 1820 GOTO 1500
K 1830 REM *** MOVE THE CURSOR
***
E 1840 IF ROW=1 THEN ROW=21
P 1850 IF COLUMN=1 THEN COLUMN=
55
E 1860 ROW=ROW-1:COLUMN=COLUM
N-1:GOTO 1570
E 1870 IF ROW=1 THEN ROW=21
N 1880 IF COLUMN=54 THEN COLUMN
=0
O 1890 ROW=ROW+1:COLUMN=COLUM
N+1:GOTO 1570
U 1900 IF ROW=20 THEN ROW=0
F 1910 IF COLUMN=1 THEN COLUMN=
55
E 1920 ROW=ROW+1:COLUMN=COLUM
N+1:GOTO 1570
M 1930 IF ROW=20 THEN ROW=0

```

```

L 1940 IF COLUMN=54 THEN COLUMN
=0
P 1950 ROW=ROW+1:COLUMN=COLUM
N+1:GOTO 1570
E 1960 IF ROW=1 THEN ROW=21
O 1970 ROW=ROW-1:GOTO 1570
K 1980 IF COLUMN=1 THEN COLUMN=
55
M 1990 COLUMN=COLUMN-1:GOTO 157
0
J 2000 COLUMN=54 THEN COLUMN
=0
E 2010 COLUMN=COLUMN+1:GOTO 157
0
L 2020 IF ROW=20 THEN ROW=0
F 2030 ROW=ROW+1:GOTO 1570
O 2040 CURS=1:GOTO 1570
E 2050 CURS=0:GOTO 1570
O 2060 CURS=1:GOTO 1570
N 2070 REM *** CLEAR THE PICTU
RE ***
N 2080 BOSUB 1520:GOTO 1550
E 2090 REM *** SAVE THE PICTUR
E ***
P 2100 LOCATE 1,24:PRINT"WAIT..
."ON 8 GOTO 2110,2120,2
130,2140,2150,2160,2170,
2180,2190,2200,2210,2220
,2230,2240,2250,2260,227
0,2280,2290,2300
J 2110 GET(1,50)=(-54,70),AX:GDS
UB 2310:GOTO 190
U 2120 GET(1,50)=(-54,70),BX:GDS
UB 2310:GOTO 190
M 2130 GET(1,50)=(-54,70),CX:GDS
UB 2310:GOTO 190
K 2140 GET(1,50)=(-54,70),DX:GDS
UB 2310:GOTO 190
E 2150 GET(1,50)=(-54,70),EX:GDS
UB 2310:GOTO 190
O 2160 GET(1,50)=(-54,70),FX:GDS
UB 2310:GOTO 190
N 2170 GET(1,50)=(-54,70),GX:GDS
UB 2310:GOTO 190
F 2180 GET(1,50)=(-54,70),HX:GDS
UB 2310:GOTO 190
N 2190 GET(1,50)=(-54,70),IX:GDS
UB 2310:GOTO 190
E 2200 GET(1,50)=(-54,70),JX:GDS
UB 2310:GOTO 190
O 2210 GET(1,50)=(-54,70),KX:GDS
UB 2310:GOTO 190
O 2220 GET(1,50)=(-54,70),LX:GDS
UB 2310:GOTO 190
N 2230 GET(1,50)=(-54,70),MX:GDS
UB 2310:GOTO 190
E 2240 GET(1,50)=(-54,70),NX:GDS
UB 2310:GOTO 190
P 2250 GET(1,50)=(-54,70),OX:GDS
UB 2310:GOTO 190
M 2260 GET(1,50)=(-54,70),PX:GDS
UB 2310:GOTO 190
O 2270 GET(1,50)=(-54,70),QX:GDS
UB 2310:GOTO 190
E 2280 GET(1,50)=(-54,70),RX:GDS
UB 2310:GOTO 190
L 2290 GET(1,50)=(-54,70),SX:GDS
UB 2310:GOTO 190
K 2300 GET(1,50)=(-54,70),TX:GDS
UB 2310:GOTO 190
K 2310 CLS:PUT(1,10),AX:PUT(61,
10),BX:PUT(121,10),CX:PU
T(181,10),OX:PUT(241,10)
,EX:PUT(301,10),FX:PUT(3
61,10),GX:PUT(421,10),HX
:PUT(481,10),IX:PUT(541,
10),JX
O 2320 PUT(1,81),KX:PUT(61,81),
LX:PUT(121,81),MX:PUT(18
1,81),NX:PUT(241,81),O
X:PUT(301,81),PX:PUT(361,
81),QX:PUT(421,81),RX:PU
T(481,81),SX:PUT(541,81),
TX,PSET

```

```

2. 2350 RETURN
M 2360 IF INKEY$="" THEN 2340 E
LBE 620
R 2350 REM *** INSERT A BLANK
PICTURE ***
M 2360 LOCATE 18,1:PRINT"ARE YO
U SURE? THIS WILL MOVE
20 OFF THE END?"
K 2370 AS=INKEY$:IF AS="" THEN
2370 ELSE IF AS<>"Y" THEN
N LOCATE 18,1:PRINT STRI
N$(50,32):GOTO 190
IF 2380 ON NUM GOTO 2390,2400,24
10,2420,2430,2440,2450,2
460,2470,2480,2490,2500,
2510,2520,2530,2540,2550
,2560,2570,2580
M 2390 GET(1,10)-(54,30),GX
U 2400 GET(61,10)-(114,30),CX
F 2410 GET(121,10)-(174,30),OZ
U 2420 GET(181,10)-(234,30),EX
F 2430 GET(241,10)-(294,30),FX
U 2440 GET(301,10)-(354,30),HX
M 2450 GET(361,10)-(414,30),HX
U 2460 GET(421,10)-(474,30),IX
M 2470 GET(481,10)-(534,30),JX
U 2480 GET(541,10)-(594,30),KX
F 2490 GET(61,81)-(114,101),LX
F 2500 GET(61,81)-(114,101),MX
U 2510 GET(121,81)-(174,101),NX
M 2520 GET(181,81)-(234,101),OZ
M 2530 GET(241,81)-(294,101),PZ
U 2540 GET(301,81)-(354,101),QZ
U 2550 GET(361,81)-(414,101),RZ
M 2560 GET(421,81)-(474,101),SX
U 2570 GET(481,81)-(534,101),TX
M 2580 CLS:ON NUM GOTO 2590,260
0,2610,2620,2630,2640,26
50,2660,2670,2680,2690,2
700,2710,2720,2730,2740,
2750,2760,2770,2780
U 2590 GET(1,50)-(54,70),AX:GOS
UB 2310:GOTO 190
U 2600 GET(1,50)-(54,70),BX:GOS
UB 2310:GOTO 190
K 2610 GET(1,50)-(54,70),CX:GOS
UB 2310:GOTO 190
M 2620 GET(1,50)-(54,70),OZ:GOS
UB 2310:GOTO 190
M 2630 GET(1,50)-(54,70),EZ:GOS
UB 2310:GOTO 190
M 2640 GET(1,50)-(54,70),FZ:GOS
UB 2310:GOTO 190
M 2650 GET(1,50)-(54,70),GZ:GOS
UB 2310:GOTO 190
F 2660 GET(1,50)-(54,70),HX:GOS
UB 2310:GOTO 190
M 2670 GET(1,50)-(54,70),IZ:GOS
UB 2310:GOTO 190
J 2680 GET(1,50)-(54,70),JZ:GOS
UB 2310:GOTO 190
U 2690 GET(1,50)-(54,70),KX:GOS
UB 2310:GOTO 190
U 2700 GET(1,50)-(54,70),LZ:GOS
UB 2310:GOTO 190
M 2710 GET(1,50)-(54,70),MZ:GOS
UB 2310:GOTO 190
U 2720 GET(1,50)-(54,70),NX:GOS
UB 2310:GOTO 190
F 2730 GET(1,50)-(54,70),OZ:GOS
UB 2310:GOTO 190
M 2740 GET(1,50)-(54,70),PX:GOS
UB 2310:GOTO 190
M 2750 GET(1,50)-(54,70),QZ:GOS
UB 2310:GOTO 190
U 2760 GET(1,50)-(54,70),RX:GOS
UB 2310:GOTO 190
U 2770 GET(1,50)-(54,70),SX:GOS
UB 2310:GOTO 190
U 2780 GET(1,50)-(54,70),TX:GOS
UB 2310:GOTO 190
M 2790 REM *** DELETE A PICTUR
E ***
M 2800 LOCATE 18,1:PRINT"ARE YO
U SURE YOU WANT TO DELE
E THIS NUMBER"
IF 2810 AS=INKEY$:IF AS="" THEN
2810 ELSE IF AS<>"Y" THEN
N LOCATE 18,1:PRINT STRI
N$(50,32):GOTO 190
M 2820 ON NUM GOTO 2830,2840,28
50,2860,2870,2880,2890,2
900,2910,2920,2930,2940,
2950,2960,2970,2980,2990
,3000,3010,3020
J 2830 GET(61,10)-(114,30),AX
M 2840 GET(121,10)-(174,30),BX
U 2850 GET(181,10)-(234,30),CX
M 2860 GET(241,10)-(294,30),OZ
U 2870 GET(301,10)-(354,30),EX
U 2880 GET(361,10)-(414,30),FX
O 2890 GET(421,10)-(474,30),GX
U 2900 GET(481,10)-(534,30),HX
J 2910 GET(541,10)-(594,30),IZ
U 2920 GET(61,81)-(114,101),JX
U 2930 GET(61,81)-(114,101),KX
M 2940 GET(121,81)-(174,101),LX
M 2950 GET(181,81)-(234,101),MX
U 2960 GET(241,81)-(294,101),NX
M 2970 GET(301,81)-(354,101),OZ
U 2980 GET(361,81)-(414,101),PX
M 2990 GET(421,81)-(474,101),QZ
K 3000 GET(481,81)-(534,101),RZ
U 3010 GET(541,81)-(594,101),SX
J 3020 CLS:GET(1,10)-(54,30),TX
:GOSUB 2310:GOTO 190
F 3030 REM *** INVERSE ALL THE
PICTURES ***
IF 3040 CLS:PUT(1,10),AX,PRESET:
PUT(61,10),BX,PRESET:PUT(1
121,10),CX,PRESET:PUT(1
81,10),OZ,PRESET:PUT(241
10),EX,PRESET:PUT(301,10
0),FX,PRESET:PUT(361,10
0),GX,PRESET:PUT(421,10),H
X,PRESET:PUT(481,10),IX,
PRESET:PUT(541,10),JX,PR
EBET
M 3050 PUT(1,81),KX,PRESET:PUT(
61,81),LX,PRESET:PUT(121
81),MX,PRESET:PUT(181,8
1),NX,PRESET:PUT(241,81),
OZ,PRESET:PUT(301,81),P
X,PRESET:PUT(361,81),QZ,
PRESET:PUT(421,81),RX,PR
EBET:PUT(481,81),SX,PRE
SET:PUT(541,81),TX,PREB
ET
M 3060 GOTO 190
IF 3070 REM *** MAKE A PROGRAM
***
M 3080 LOCATE 18,1:PRINT"ARE YO
U SURE YOU WANT TO MAKE
THIS SET-UP INTO A PROGR
AM?"
J 3090 AS=INKEY$:IF AS="" THEN
3090 ELSE IF AS="" THEN
3100 ELSE LOCATE 18,1:P
RINT STRIN$(66,32):GOTO
290
M 3100 G=30:OPEN "O",#1,"PRG.BA
B"
J 3110 PRINT #1,"10 CLS:KEY OFF
:SCREEN 2:SP=+STR$(SP)+
"PL=1"
J 3120 IF STAYEN THEN GO--1 ELS
E GO=1
IF 3130 AS="29 OIM "G0=1:STA T
O EN STEP:G0=1:IF I<STA T
HEN AS="0"
M 3140 ON I GOTO 3150,3160,3170
,3180,3190,3200,3210,322
0,3230,3240,3250,3260,3
270,3280,3290,3300,3310,3
320,3330,3340
M 3150 AS=AS+"A2(144)":GOTO 335
0
M 3160 AS=AS+"BX(144)":GOTO 335
0
M 3170 AS=AS+"CX(144)":GOTO 335
0
E 3180 AS=AS+"OZ(144)":GOTO 335
0
F 3190 AS=AS+"EZ(144)":GOTO 335
0
A 3200 AS=AS+"FX(144)":GOTO 335
0
F 3210 AS=AS+"GX(144)":GOTO 335
0
A 3220 AS=AS+"HX(144)":GOTO 335
0
M 3230 AS=AS+"IZ(144)":GOTO 335
0
A 3240 AS=AS+"JZ(144)":GOTO 335
0
J 3250 AS=AS+"KZ(144)":GOTO 335
0
A 3260 AS=AS+"LX(144)":GOTO 335
0
U 3270 AS=AS+"MX(144)":GOTO 335
0
M 3280 AS=AS+"NX(144)":GOTO 335
0
M 3290 AS=AS+"OZ(144)":GOTO 335
0
M 3300 AS=AS+"PX(144)":GOTO 335
0
E 3310 AS=AS+"QZ(144)":GOTO 335
0
E 3320 AS=AS+"RX(144)":GOTO 335
0
F 3330 AS=AS+"SX(144)":GOTO 335
0
E 3340 AS=AS+"TX(144)":GOTO 335
0
M 3350 NEXT
E 3360 AS=AS+",UX(144)":PRINT #
1,AS:PRINT #1,"30 GET(1,
1)-(54,20),UX"
M 3370 IF STAYEN THEN GO--1 ELS
E GO=1
M 3380 FOR P=STA TO EN STEP GO
E 3390 CLS:ON P GOTO 3400,3410,
3420,3430,3440,3450,3460
,3470,3480,3490,3500,351
0,3520,3530,3540,3550,35
60,3570,3580,3590
F 3400 PUT(1,50),AX:GOTO 3600
R 3410 PUT(1,50),BX:GOTO 3600
R 3420 PUT(1,50),CX:GOTO 3600
O 3430 PUT(1,50),OZ:GOTO 3600
M 3440 PUT(1,50),EZ:GOTO 3600
E 3450 PUT(1,50),FX:GOTO 3600
E 3460 PUT(1,50),GX:GOTO 3600
M 3470 PUT(1,50),HX:GOTO 3600
L 3480 PUT(1,50),IZ:GOTO 3600
J 3490 PUT(1,50),JZ:GOTO 3600
K 3500 PUT(1,50),KX:GOTO 3600
M 3510 PUT(1,50),LZ:GOTO 3600
A 3520 PUT(1,50),MX:GOTO 3600
F 3530 PUT(1,50),NX:GOTO 3600
M 3540 PUT(1,50),OZ:GOTO 3600
F 3550 PUT(1,50),PX:GOTO 3600
E 3560 PUT(1,50),QZ:GOTO 3600
F 3570 PUT(1,50),RX:GOTO 3600
O 3580 PUT(1,50),SX:GOTO 3600
F 3590 PUT(1,50),TX:GOTO 3600
M 3600 FOR I=1 TO 54:G=0:IF AS=
STR$(G):AS=RIGHT$(AS,LEN
(AS)-1):FOR I=1 TO 20:IF
POINT(I,1+49)=1 THEN AS
=AS+"PSET("AS=STR$(G):
AS=AS+RIGHT$(8,LEN(AS)-
1):AS=AS+"",80=STR$(I+4
9):AS=AS+RIGHT$(8,LEN(B
9)-1):AS=AS+"")
F 3610 NEXT:IF LEN(AS)>6 THEN B
=LEFT$(AS,LEN(STR$(B))-
1):AS=RIGHT$(AS,LEN(AS)-
LEN(B))-1:AS=AS+B+" "+A
:PRINT #1,AS ELSE G=0-I
M 3620 NEXT

```

```

10 3630 G=0+10: AS=STR$(G): A$=RIG
HT$(A$,LEN(A$)-1): ON P G
OTO 3640,3650,3660,3670,
3680,3690,3700,3710,3720,
3730,3740,3750,3760,377
0,3780,3790,3800,3810,38
20,3830
M 3640 AS=A$+" GET(1,50)-(54,70
),A$:I$OTO 3840
P 3650 AS=A$+" GET(1,50)-(54,70
),B$:I$OTO 3840
C 3660 AS=A$+" GET(1,50)-(54,70
),C$:I$OTO 3840
F 3670 AS=A$+" GET(1,50)-(54,70
),D$:I$OTO 3840
J 3680 AS=A$+" GET(1,50)-(54,70
),E$:I$OTO 3840
L 3690 AS=A$+" GET(1,50)-(54,70
),F$:I$OTO 3840
N 3700 AS=A$+" GET(1,50)-(54,70
),G$:I$OTO 3840
P 3710 AS=A$+" GET(1,50)-(54,70
),H$:I$OTO 3840
R 3720 AS=A$+" GET(1,50)-(54,70
),I$:I$OTO 3840
Q 3730 AS=A$+" GET(1,50)-(54,70
),J$:I$OTO 3840
S 3740 AS=A$+" GET(1,50)-(54,70
),K$:I$OTO 3840
U 3750 AS=A$+" GET(1,50)-(54,70
),L$:I$OTO 3840
W 3760 AS=A$+" GET(1,50)-(54,70
),M$:I$OTO 3840
Y 3770 AS=A$+" GET(1,50)-(54,70
),N$:I$OTO 3840
A 3780 AS=A$+" GET(1,50)-(54,70
),O$:I$OTO 3840
C 3790 AS=A$+" GET(1,50)-(54,70
),P$:I$OTO 3840
E 3800 AS=A$+" GET(1,50)-(54,70
),Q$:I$OTO 3840
G 3810 AS=A$+" GET(1,50)-(54,70
),R$:I$OTO 3840
I 3820 AS=A$+" GET(1,50)-(54,70
),S$:I$OTO 3840
K 3830 AS=A$+" GET(1,50)-(54,70
),T$:I$OTO 3840
M 3840 AS=A$+"CLS":PRINT #1,A$
NEXT
K 3850 IF STA=END THEN STN=1:EA=
1:I$OTO 3880
M 3860 IF STA=END THEN STN=STA+E
N:EA=1 ELSE EA=EN=STA+ST
N:EA=1
O 3870 IF EA=QD=0 THEN EA=EQ-QD
O 3880 G=0+10:R=0:AS=RIGHT$(STR
$(G),LEN(STR$(G))-1)+": F
OR I=+RIGHT$(STR$(STN),
LEN(STR$(STN))-1)+": TO+
STR$(EA-QD)+"STEP" *STR$
(QD):PRINT #1,A$:G=0+10
M 3890 AS=RIGHT$(STR$(G),LEN(STR
$(G))-1)+": FOR P=1 TO:
AS=AS+STR$(INT(FAST4.2)
)+":NEXT:PL=PL+I:IF PL=0P
<1 THEN PL=578 ELSE IF P
L=SP>580 THEN PL=1
N 3900 G=0+10:PRINT #1,A$
P 3910 AS=RIGHT$(STR$(G),LEN(STR
$(G))-1)+": PL=PL+SP+P:PR
INT #1,A$:G=0+10
X 3920 AS=RIGHT$(STR$(G),LEN(STR
$(G))-1)+": ON I:OTO:K
=0+10:AS=AS+STR$(X):FOR
I=STA TO EN:QD=1:STEP
QD:X=X+10:AS=AS+"*RIG
HT$(STR$(X),LEN(STR$(X)
)-1):NEXT:PRINT #1,A$
M 3930 IF STA=END THEN Q=STA+M
N ELSE IF EN=STA THEN Q=
EN+M=STA
P 3940 FOR P=M TO Q
P 3950 G=0+10:AS=RIGHT$(STR$(G)
,LEN(STR$(G))-1):ON P:GO
TO 3960,3970,3980,3990,4
000,4010,4020,4030,4040,
4050,4060,4070,4080,4090
,4100,4110,4120,4130,414
0,4150
I 3960 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),A$:I$OTO
4160
L 3970 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),B$:I$OTO
4160
O 3980 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),C$:I$OTO
4160
E 3990 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),D$:I$OTO
4160
F 4000 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),E$:I$OTO
4160
H 4010 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),F$:I$OTO
4160
J 4020 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),G$:I$OTO
4160
L 4030 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),H$:I$OTO
4160
N 4040 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),I$:I$OTO
4160
P 4050 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),J$:I$OTO
4160
R 4060 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),K$:I$OTO
4160
T 4070 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),L$:I$OTO
4160
V 4080 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),M$:I$OTO
4160
X 4090 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),N$:I$OTO
4160
Z 4100 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),O$:I$OTO
4160
B 4110 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),P$:I$OTO
4160
D 4120 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),Q$:I$OTO
4160
F 4130 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),R$:I$OTO
4160
H 4140 AS=A$+" PUT(PL,52),U$,P
SET:PUT(PL,52),S$:I$OTO
4160
J 4150 AS="GOTO"+STR$(X+10):
PRINT #1,A$:NEXT:G=X+10
L 4170 AS=RIGHT$(STR$(G),LEN(STR
$(G))-1)+":AS=AS+"NEXT:G
OTO"+STR$(R):PRINT #1,A$
N 4180 CLOSE #1:CLS:PRINT"BEFOR
E YOU DO ANYTHING ELSE I
QAD THE PROGRAM (PRG) AN
O THEN SAVE IT UNDER THE
NAME YOU WANT":END
O 250000 GOSUB 25020:RESUME 990
M 25010 GOSUB 25020:RESUME 870
P 25020 PRINT:PRINT "
K ERROR %"ERR
P 25030 PRINT:PRINT "
HIT SPACE
BAR TO CONTINUE"

```

```

O 25040 AS=INKEY$: IF
EN 25050 ELSE:

```

## Program 2: 64 Animator

Version by Kevin Mykytyn, Editorial Programmer

Please refer to the "ALX" article in this issue before entering the following listing.

```

49152 1032,238,283,032,142,283,002
49155 1608,127,105,123,286,153,192
49164 064,083,213,016,247,076,042
49170 046,192,000,000,001,001,015,017
49176 000,1,01,000,000,000,000,045
49182 000,000,000,000,000,000,000,070
49188 000,000,000,000,000,000,000,042
49194 000,000,000,000,000,000,000,042
49200 192,169,000,141,020,192,002
49206 032,070,193,032,044,193,106
49212 032,078,194,032,215,193,208
49218 032,033,197,162,255,032,009
49224 205,194,076,000,192,120,151
49230 169,127,141,013,220,169,149
49236 105,141,020,000,169,192,002
49242 141,021,000,169,129,141,012
49248 026,201,100,027,141,017,152
49254 000,000,000,169,001,141,037
49260 025,200,173,044,192,200,100
49266 069,173,018,200,201,010,025
49272 176,011,162,225,160,059,145
49278 169,003,141,010,200,200,105
49284 029,200,005,176,011,162,208
49290 232,160,009,169,130,141,045
49296 010,200,200,014,162,239,225
49302 168,139,169,000,141,010,009
49308 200,240,000,140,010,200,205
49314 152,160,012,153,001,200,000
49320 136,136,016,249,160,000,097
49326 138,153,240,007,200,232,120
49332 192,007,200,246,173,013,251
49338 220,041,001,240,001,164,165
49344 203,204,043,192,240,056,100
49350 140,001,192,105,129,235,000
49356 201,000,200,000,171,029,140
49362 200,073,120,141,020,200,229
49368 201,009,200,000,171,023,150
49374 200,073,120,141,023,200,235
49380 201,049,144,022,201,057,134
49386 176,010,056,174,141,002,033
49392 242,004,200,010,233,049,200
49398 160,007,153,030,200,136,101
49404 166,250,201,133,200,011,047
49410 171,005,192,240,000,032,170
49416 019,193,032,230,203,076,001
49422 049,234,076,180,254,173,220
49428 136,002,073,012,141,136,000
49434 002,173,024,200,073,040,042
49440 141,024,200,173,021,200,039
49446 073,127,141,021,200,096,192
49452 169,127,141,021,200,169,112
49458 004,141,016,200,160,006,133
49464 012,012,105,047,204,157,055
49470 000,200,202,002,136,016,050
49476 245,096,169,147,032,210,199
49482 255,169,015,141,031,200,127
49488 162,001,160,002,024,032,205
49494 240,255,032,197,193,162,141
49500 000,160,002,024,032,240,044
49506 255,032,197,193,162,011,100
49512 160,002,024,032,240,255,049
49518 032,197,193,162,000,160,090
49524 240,032,240,255,169,070
49530 005,160,240,032,030,171,036
49536 162,009,160,002,024,032,005
49542 240,255,169,120,160,204,002
49548 032,030,171,162,014,160,197
49554 002,032,032,240,255,160,000
49560 155,160,004,032,240,171,136
49566 032,165,193,032,101,193,106
49572 009,162,019,160,001,024,114
49578 032,240,255,169,160,160,192
49584 204,032,030,171,096,162,103
49590 032,160,001,024,032,240,149
49596 255,169,116,160,205,032,101
49602 030,171,096,169,007,141,040
49608 030,169,169,030,160,204,231
49614 032,030,171,200,020,192,009
49620 200,244,096,162,019,160,077
49626 010,024,032,240,255,169,100

```

49632 0008,174,021,192,832,061,192  
49638 194,162,019,160,838,024,459  
49644 0032,240,255,169,080,174,082  
49650 0024,192,832,061,194,162,139  
49656 0020,160,818,024,032,240,238  
49662 255,169,080,174,822,192,042  
49668 0032,061,194,032,164,136,171  
49674 162,821,160,018,024,032,171  
49680 240,255,169,080,174,082,139  
49686 192,082,032,061,194,162,091  
49692 0021,160,838,024,832,240,837  
49698 255,169,080,174,822,192,082  
49704 224,016,144,082,162,015,091  
49710 142,838,192,032,061,194,191  
49716 173,838,192,080,816,141,109  
49722 012,212,096,32,218,255,089,036  
49728 169,832,032,218,255,089,036  
49734 174,820,192,173,080,220,089  
49740 074,176,814,082,159,194,213  
49746 202,816,082,162,086,032,246  
49752 159,194,076,154,194,074,131  
49758 176,816,032,159,194,232,175  
49764 224,087,144,082,162,081,128  
49770 0032,159,194,076,154,194,147  
49776 074,176,814,189,020,192,088  
49782 240,255,169,080,174,082,139  
49788 192,076,154,194,076,154,194  
49794 013,189,020,192,081,021,254  
49800 240,083,254,028,192,076,153  
49806 154,194,074,176,087,138,117  
49812 0072,032,225,194,104,170,177  
49818 142,082,192,096,096,072,812  
49824 130,872,018,170,189,051,822  
49830 285,133,251,189,052,205,177  
49836 133,252,160,815,177,251,136  
49842 0073,128,145,251,136,016,156  
49848 247,184,170,189,096,169,089  
49854 0081,141,028,192,169,080,209  
49860 160,084,153,021,192,136,094  
49866 816,250,096,160,188,136,192  
49872 208,251,202,280,248,096,143  
49878 832,285,194,173,080,220,014  
49884 041,816,240,246,096,032,123  
49890 214,194,169,255,141,045,228  
49896 192,141,021,280,192,169,089  
49902 194,189,192,169,080,141,066  
49908 015,2087,173,036,191,141,241  
49914 046,280,173,024,192,285,074  
49920 021,192,176,083,238,029,147  
49926 192,173,021,192,141,022,235  
49932 192,169,080,141,037,192,231  
49938 169,021,056,327,823,192,084  
49944 141,832,192,081,021,208,051  
49950 0803,230,327,192,141,025,166  
49956 192,141,034,192,173,032,182  
49962 192,240,028,173,037,192,130  
49968 208,113,206,033,192,208,248  
49974 108,169,180,141,833,192,029  
49980 206,832,192,208,098,169,197  
49986 021,056,237,032,192,141,224  
49992 832,192,173,021,192,285,119  
49998 024,192,248,041,173,022,082  
50004 192,208,244,192,208,082,145  
50010 173,821,192,161,032,192,063  
50016 208,075,173,029,182,240,185  
50022 011,173,022,192,056,233,021  
50028 0081,141,022,192,080,099,169  
50034 173,822,192,024,105,081,119  
50040 141,022,192,168,105,228,024  
50046 055,240,018,169,120,141,181  
50052 004,212,169,129,141,084,023  
50058 212,169,080,173,032,192,075  
50064 141,080,212,185,024,056,082  
50070 141,087,212,169,032,141,084  
50076 011,212,169,033,141,011,221  
50082 212,169,080,141,838,192,130  
50088 173,834,192,240,085,206,250  
50094 834,192,208,060,173,025,098  
50100 192,056,233,011,016,080,186  
50106 230,816,192,074,235,016,186  
50112 105,081,141,034,192,173,070  
50118 034,192,176,240,087,024,097  
50124 109,834,192,202,208,249,174  
50130 141,035,192,173,038,192,205  
50136 240,088,173,031,192,056,188  
50142 237,035,192,141,031,192,026  
50148 173,814,208,233,080,142,026  
50154 014,208,233,251,169,080,186  
50160 173,816,208,061,220,208,246  
50166 015,169,083,141,814,208,110  
50172 173,816,208,089,034,141,159



"64 Animator" uses raster interrupt techniques to display up to 22 sprites at once.

50178 016,208,200,060,173,816,179  
50184 208,041,127,141,016,208,237  
50190 206,014,208,076,074,196,028  
50196 173,813,192,024,189,035,072  
50202 192,141,034,192,173,014,061  
50208 208,105,080,141,014,208,196  
50214 208,010,173,016,208,099,158  
50220 120,141,016,208,208,024,001  
50226 208,005,208,028,173,016,241  
50232 208,041,120,248,013,169,087  
50238 018,141,014,208,173,016,112  
50244 208,041,127,141,016,208,041  
50250 234,173,080,238,074,074,001  
50256 074,176,814,173,025,192,224  
50262 248,054,206,027,192,208,245  
50268 049,206,025,192,076,142,014  
50274 196,074,176,814,173,025,240  
50280 192,201,021,240,033,230,005  
50286 027,192,208,028,238,025,060  
50292 192,076,142,196,074,176,284  
50298 199,169,080,141,045,192,176  
50304 173,136,082,280,040,248,116  
50310 005,032,192,192,192,089,089  
50316 194,096,032,164,196,032,086  
50322 235,196,173,022,192,168,108  
50328 204,085,224,141,255,087,140  
50334 141,255,011,076,048,195,188  
50340 169,080,141,069,219,141,135  
50346 028,219,141,071,219,173,039  
50352 072,192,056,233,011,072,231  
50358 016,014,169,083,192,169,083  
50364 087,184,073,255,024,185,244  
50370 081,076,204,196,169,032,184  
50376 141,069,087,104,281,018,220  
50382 144,016,077,169,049,141,029  
50388 087,087,104,056,233,010,180  
50394 089,048,141,071,087,096,078  
50400 089,048,141,070,087,169,156  
50406 032,141,071,087,096,173,230  
50412 022,192,081,028,144,016,016  
50418 072,169,050,141,050,087,219  
50424 104,056,233,028,089,048,208  
50430 141,051,087,096,201,018,240  
50436 144,016,077,169,049,141,083  
50442 050,087,104,056,233,010,214  
50448 089,048,141,051,087,096,112  
50454 089,048,141,050,087,169,130  
50460 032,141,051,087,096,032,131  
50466 228,255,201,069,208,083,238  
50472 076,184,199,201,067,189,207  
50478 038,032,082,204,176,033,019  
50484 169,064,133,251,169,056,126  
50490 133,252,032,126,203,190,234  
50496 252,032,126,203,168,021,098  
50502 169,080,153,220,055,153,052  
50508 258,055,153,024,056,136,238  
50514 016,244,096,201,073,208,152  
50520 096,169,120,133,251,169,018  
50526 061,133,252,169,064,133,138  
50532 253,169,061,133,254,169,133  
50538 022,056,327,022,192,170,087  
50544 240,071,160,063,177,253,052  
50550 145,251,136,016,249,105,056  
50556 251,056,233,064,133,251,080  
50562 165,252,233,000,133,252,141  
50568 253,165,233,000,133,252,141  
50574 253,165,234,233,000,133,156  
50580 254,282,208,218,162,020,188  
50586 199,250,055,072,109,024,165  
50592 056,072,109,220,055,232,216

50598 157,220,055,104,157,024,115  
50604 056,184,157,250,055,202,228  
50610 202,236,022,192,176,226,088  
50616 006,201,068,208,173,173,008  
50622 022,192,832,165,199,165,197  
50628 251,024,105,064,133,253,002  
50634 165,252,105,080,133,254,087  
50640 169,022,056,237,022,192,138  
50646 170,240,072,160,063,177,072  
50652 253,145,251,136,016,249,246  
50658 165,252,105,064,133,253,008  
50664 251,165,252,105,080,133,114  
50670 252,165,253,024,105,064,077  
50676 133,253,165,254,105,080,130  
50682 133,254,202,016,230,174,223  
50688 022,192,232,189,250,055,172  
50694 072,189,024,056,072,109,096  
50700 220,055,202,157,230,055,153  
50706 184,157,024,056,184,157,108  
50712 250,055,232,232,224,021,014  
50718 144,227,096,201,083,208,221  
50724 044,032,146,199,032,063,040  
50730 199,169,208,133,253,169,161  
50736 055,133,254,173,024,192,111  
50742 032,165,199,165,251,024,122  
50748 105,064,170,165,252,105,153  
50754 080,168,169,253,032,216,136  
50760 076,046,192,201,076,208,109  
50766 076,046,192,201,076,208,109  
50772 019,032,146,199,032,063,063  
50778 199,169,080,032,213,255,190  
50784 032,233,190,184,104,076,075  
50790 046,192,201,036,208,099,116  
50796 032,146,199,169,081,162,048  
50802 080,160,080,032,196,255,133  
50808 169,080,162,059,160,208,116  
50814 032,189,255,032,192,255,016  
50820 032,233,198,162,081,032,022  
50826 198,255,032,207,255,032,093  
50832 207,255,169,031,032,210,086  
50838 255,032,207,255,032,207,114  
50844 255,032,233,198,176,032,058  
50850 032,207,255,170,032,207,041  
50856 255,032,205,169,032,169,032,026  
50862 032,210,207,255,169,032,026  
50868 240,230,032,210,255,165,022  
50874 203,081,064,208,250,076,164  
50880 177,198,169,081,032,195,196  
50886 255,032,204,255,104,104,128  
50892 076,046,192,201,081,208,240  
50898 006,032,082,204,176,080,119  
50904 080,096,169,080,168,206,079  
50910 032,210,207,255,169,032,026  
50916 081,031,208,240,096,141,014  
50922 030,192,024,165,144,240,013  
50928 074,041,120,248,028,169,144  
50934 099,160,206,032,038,171,176  
50940 169,081,032,195,255,032,168  
50946 218,198,184,104,076,202,136  
50952 199,169,015,162,080,168,216  
50958 032,186,055,192,171,171,030  
50964 080,072,192,255,169,032,026  
50970 015,032,198,255,169,031,196  
50976 032,210,255,032,207,255,255  
50982 032,210,255,169,031,196  
50988 013,208,244,169,015,032,213  
50994 195,255,032,204,255,032,255  
51000 218,199,056,173,030,192,163  
51006 096,169,218,160,203,032,166  
51012 080,271,032,230,171,171,030  
51018 080,080,032,195,184,104,241  
51024 076,076,188,168,255,208,021  
51030 115,080,002,153,030,002,218  
51036 208,247,162,005,189,060,195  
51042 206,157,032,082,208,202,129  
51048 016,246,152,162,032,160,184  
51054 002,032,189,255,169,229,218  
51060 032,210,255,169,031,196  
51066 220,255,081,084,208,034,070  
51072 162,081,208,080,201,060,086  
51078 200,241,162,080,169,082,156  
51084 160,081,032,186,255,086,102  
51090 032,091,255,032,138,255,181  
51096 169,024,141,024,208,169,119  
51102 251,160,205,032,038,171,239  
51108 096,024,105,244,133,251,239  
51114 169,080,133,252,162,208,234  
51120 096,253,038,252,202,208,199  
51126 249,096,169,081,141,044,114  
51132 192,169,055,141,011,208,196  
51138 141,013,208,169,035,141,133

51144 .012,208,169,255,141,010,227  
51150 .208,169,224,141,021,208,153  
51156 .169,195,141,016,208,169,086  
51162 .020,141,014,208,169,005,007  
51168 .141,015,208,169,084,133,266  
51174 .247,169,084,141,046,169,176  
51180 .020,141,000,208,141,002,236  
51186 .208,169,165,141,001,208,110  
51192 .169,190,141,003,208,169,104  
51198 .013,141,248,007,169,014,078  
51204 .141,249,007,032,170,282,037  
51210 .172,022,192,185,220,055,000  
51216 .240,020,173,021,208,009,175  
51222 .001,141,021,208,169,120,178  
51228 .141,004,212,169,129,141,005  
51234 .000,212,208,173,021,140,140  
51240 .208,001,254,141,021,208,145  
51246 .172,022,192,185,250,055,154  
51252 .240,038,141,008,212,105,100  
51258 .024,056,141,007,212,169,155  
51264 .032,141,011,212,173,063,150  
51270 .141,011,212,173,021,208,068  
51276 .009,002,141,021,208,076,021  
51282 .092,208,173,021,208,041,049  
51288 .253,141,021,208,007,194,165  
51294 .208,212,187,169,009,134  
51300 .133,253,133,141,169,216,121  
51306 .133,253,133,142,173,022,195  
51312 .192,024,105,224,141,255,029  
51318 .007,056,233,001,141,253,041  
51324 .007,240,105,002,141,254,145  
51330 .007,173,022,192,032,165,209  
51336 .199,165,251,133,141,165,164  
51342 .252,133,140,169,008,141,209  
51348 .030,192,169,003,141,039,210  
51354 .192,172,038,192,177,251,152  
51360 .140,038,192,168,008,162,084  
51366 .000,010,144,006,072,169,063  
51372 .001,145,253,140,230,253,134  
51378 .208,002,230,254,020,008,002  
51384 .230,230,038,192,173,030,077  
51390 .192,208,064,201,021,208,009  
51396 .039,192,208,111,165,251,240  
51402 .024,105,016,133,253,165,130  
51408 .254,105,000,133,254,076,006  
51414 .150,208,165,247,133,249,070  
51420 .165,248,133,250,160,008,152  
51426 .177,247,073,120,145,247,219  
51432 .162,120,032,205,194,073,002  
51438 .120,145,247,010,008,220,127  
51444 .074,176,015,172,165,247,005  
51450 .056,233,040,133,247,165,100  
51456 .240,233,000,133,240,184,190  
51462 .074,176,015,072,165,247,243  
51468 .240,185,040,133,247,165,214  
51474 .246,105,000,133,240,184,008  
51480 .074,176,015,072,165,247,005  
51486 .056,233,001,133,247,165,097  
51492 .240,233,000,133,240,184,234  
51498 .074,176,015,072,165,247,023  
51504 .024,105,001,133,247,165,211  
51510 .240,185,000,133,240,184,124  
51516 .074,176,030,165,249,024,010  
51522 .105,000,133,070,165,250,029  
51528 .105,212,133,079,173,061,047  
51534 .192,208,012,169,001,141,033  
51540 .041,192,177,078,073,001,134  
51546 .141,040,192,173,040,192,100  
51552 .145,078,076,106,221,169,103  
51558 .000,141,011,212,247,132  
51564 .001,032,208,008,165,249,203  
51570 .133,247,165,250,133,240,010  
51576 .032,228,255,201,002,208,102  
51582 .013,169,000,141,044,192,173  
51588 .169,129,141,026,208,076,113  
51594 .009,192,201,147,208,003,170  
51600 .032,170,202,201,008,208,232  
51606 .013,173,022,192,021,001,009  
51612 .240,006,230,022,192,076,162  
51618 .010,208,201,045,208,013,071  
51624 .173,022,192,201,001,240,229  
51630 .006,206,022,192,076,010,174  
51636 .208,201,073,200,014,160,012  
51642 .063,177,139,073,255,145,014  
51648 .139,136,016,247,076,092,130  
51654 .200,001,136,247,165,160,009  
51660 .063,177,139,153,122,206,040  
51666 .136,016,248,201,140,208,135  
51672 .013,160,063,185,122,206,197  
51678 .145,139,136,016,240,076,214  
51684 .092,208,201,145,208,011,061  
51690 .162,003,032,001,203,202,129  
51696 .208,250,076,092,200,201,243  
51702 .017,208,001,162,003,032,167  
51708 .091,203,202,200,250,076,002  
51714 .092,208,001,020,200,037,247  
51720 .101,002,177,139,106,167,247  
51726 .136,162,003,177,139,106,225  
51732 .145,139,208,202,200,240,137  
51738 .208,200,192,064,144,234,030  
51744 .076,092,200,201,157,200,190  
51750 .029,160,008,177,139,042,073  
51756 .208,200,162,003,177,139,157  
51762 .042,145,139,136,202,200,154  
51768 .067,200,200,200,200,030,015  
51774 .000,144,232,076,092,000,000  
51780 .001,060,200,173,172,022,241  
51786 .192,185,220,055,073,001,032  
51792 .153,220,055,076,010,200,026  
51798 .173,141,002,201,004,240,079  
51804 .062,164,203,105,129,235,066  
51810 .205,060,003,240,052,141,031  
51816 .060,003,201,049,144,045,094  
51822 .001,050,176,041,174,022,044  
51828 .192,056,233,049,160,105,231  
51834 .050,205,157,250,055,105,001  
51840 .041,206,157,024,056,173,017  
51846 .141,002,240,014,109,024,232  
51852 .056,010,157,024,056,189,120  
51858 .250,055,042,157,250,055,107  
51864 .076,010,200,201,092,200,191  
51870 .003,076,010,200,032,225,192  
51876 .202,160,000,076,216,200,250  
51882 .169,065,160,055,032,080,003  
51888 .171,169,021,141,026,192,122  
51894 .169,060,160,205,032,030,070  
51900 .171,206,020,192,200,244,205  
51906 .162,012,160,033,024,032,105  
51912 .240,255,169,000,174,022,030  
51918 .192,032,205,189,169,032,001  
51924 .032,210,255,096,160,008,197  
51930 .177,139,041,137,145,139,218  
51936 .096,165,141,137,163,165,063  
51942 .192,032,205,189,169,032,001  
51948 .030,192,169,003,141,039,100  
51954 .192,160,007,169,008,141,143  
51960 .042,192,024,177,163,041,119  
51966 .015,240,001,056,110,042,206  
51972 .192,136,016,242,172,030,032  
51978 .192,173,042,192,145,139,125  
51984 .165,163,042,185,008,133,102  
51990 .165,164,164,105,008,133,148  
51996 .164,300,030,192,169,000,000  
52002 .192,201,064,200,005,096,020  
52008 .200,039,192,200,190,165,024  
52014 .163,024,105,016,133,163,130  
52020 .165,164,105,000,133,164,010  
52026 .170,230,202,160,000,177,143  
52032 .139,141,030,192,136,200,142  
52038 .200,192,064,200,008,173,147  
52044 .030,192,136,136,145,139,094  
52050 .007,177,139,136,145,139,094  
52056 .076,069,203,160,002,177,067  
52062 .139,141,030,192,200,136,167  
52068 .136,016,007,173,030,192,150  
52074 .200,145,139,096,177,139,234  
52080 .200,145,139,076,099,203,206  
52086 .169,000,133,251,169,216,032  
52092 .133,252,162,064,160,008,067  
52098 .192,145,251,160,200,251,249  
52104 .220,254,200,200,200,096,000  
52110 .120,165,001,041,251,133,005  
52116 .001,169,000,137,251,169,103  
52122 .032,133,252,169,000,133,105  
52128 .253,169,200,133,254,162,059  
52134 .000,160,000,177,253,145,141  
52140 .253,160,200,249,230,252,210  
52146 .220,254,200,200,200,096,000  
52152 .001,009,004,133,001,000,000  
52158 .160,007,105,000,205,153,232  
52164 .000,033,105,100,205,153,120  
52170 .000,037,136,016,241,169,041  
52176 .024,141,024,200,160,023,000  
52182 .169,000,153,000,212,136,116  
52188 .016,250,169,015,011,024,067  
52194 .212,160,200,141,000,141,000  
52200 .169,019,141,005,212,160,000  
52206 .169,032,160,000,153,000,200  
52212 .000,153,000,009,153,000,055  
52218 .010,153,000,011,136,200,000  
52224 .241,096,032,030,204,169,012  
52230 .067,160,206,032,030,171,160  
52236 .032,220,235,201,009,024,074  
52242 .240,250,201,070,200,244,226  
52248 .056,008,032,030,240,169,019  
52254 .086,160,206,032,030,171,200  
52260 .040,006,162,150,030,011  
52266 .024,032,240,255,096,040,225  
52272 .008,120,160,208,240,033,153  
52278 .144,102,163,163,163,181,026  
52284 .017,157,157,157,157,157,094  
52290 .102,032,032,032,181,017,030  
52296 .157,157,157,032,157,102,015  
52302 .175,175,173,181,145,145,090  
52308 .032,032,032,032,032,032,032  
52314 .032,032,032,032,032,032,032  
52320 .032,032,032,032,032,032,032  
52326 .032,032,032,032,032,032,032  
52332 .032,032,032,032,032,032,032  
52338 .032,032,032,032,032,032,032  
52344 .032,032,032,032,032,032,032  
52350 .032,032,032,032,032,032,032  
52356 .040,040,032,032,032,032,032  
52362 .040,032,032,032,032,032,032  
52368 .032,032,032,032,032,032,032  
52374 .032,032,032,032,032,032,032  
52380 .040,032,032,032,032,032,032  
52386 .040,032,032,032,032,032,032  
52392 .032,032,032,032,032,032,032  
52398 .032,032,032,032,032,032,032  
52404 .032,032,032,032,032,032,032  
52410 .032,032,032,032,032,032,032  
52416 .032,032,032,032,032,032,032  
52422 .070,071,032,060,167,007,007  
52428 .040,005,002,060,146,144,066  
52434 .032,032,032,032,032,032,032  
52440 .073,073,073,073,073,073,073  
52446 .073,073,073,073,073,073,073  
52452 .032,032,032,032,032,032,032  
52458 .067,040,085,008,067,032,141  
52464 .070,005,077,066,069,082,185  
52470 .032,032,032,032,032,032,032  
52476 .070,066,069,082,065,076,197  
52482 .076,032,083,008,069,069,155  
52488 .076,032,083,008,069,069,155  
52494 .003,007,073,004,007,072,224  
52500 .073,070,071,032,083,008,101  
52506 .069,069,068,032,032,032,032  
52512 .032,032,070,070,070,069,150  
52518 .032,068,005,082,065,084,190  
52524 .073,070,070,070,070,070,070  
52530 .040,040,040,040,040,040,040  
52536 .007,070,007,019,007,053,216  
52542 .032,032,032,032,032,032,032  
52548 .031,146,144,029,029,029,202  
52554 .029,033,033,033,033,033,033  
52560 .033,033,033,033,033,033,032  
52566 .033,033,033,033,033,033,032  
52572 .033,033,033,033,033,033,034  
52578 .033,000,000,126,126,126,253  
52584 .126,126,126,000,253,129,098  
52590 .129,129,129,129,129,253,242  
52596 .000,000,000,000,000,000,000  
52602 .073,084,032,032,032,032,155  
52608 .032,032,010,076,140,070,250  
52614 .065,068,032,032,032,032,139  
52620 .032,032,010,083,146,065,084  
52626 .006,069,032,032,032,032,173  
52632 .032,032,010,067,146,076,011  
52638 .009,065,002,032,065,076,035  
52644 .076,032,032,032,032,032,032  
52650 .000,064,065,076,077,077,064  
52656 .032,032,032,032,032,032,032  
52662 .003,009,002,004,032,032,032  
52668 .032,032,010,069,146,069,041  
52674 .076,069,004,069,032,032,032  
52680 .032,032,010,081,146,005,082  
52686 .073,004,144,000,017,017,029  
52692 .073,032,032,032,032,032,032  
52698 .032,070,073,076,069,070,064  
52704 .067,077,069,011,000,017,209  
52710 .017,032,010,084,146,065,080  
52716 .000,069,032,079,002,032,032  
52722 .010,060,146,073,008,075,193  
52728 .013,013,000,147,005,017,187  
52734 .017,000,013,017,032,072,149  
52740 .073,032,032,032,032,032,032  
52746 .005,002,070,032,004,070,134  
52752 .032,067,079,070,084,073,173  
52758 .070,005,000,000,000,002,140  
52764 .005,077,008,070,070,084,132  
52770 .009,000,032,032,032,032,231  
52776 .009,097,104,143,040,143,063



```

52702 :024,210,195,000,000,009,236
52708 :010,011,012,014,015,016,130
52714 :000,036,040,050,065,070,087
52800 :073,046,042,020,005,032,200
52806 :069,012,009,079,005,032,200
52812 :003,005,002,069,063,144,009
52818 :000,032,032,032,032,032,242
52824 :032,032,032,032,032,032,242
52830 :032,032,000,073,040,013,036
52836 :013,032,060,069,006,073,105
52842 :007,069,032,070,079,004,003
52848 :032,000,002,069,003,069,015
52854 :070,004,013,000,000,032,069
52860 :000,004,016,000,000,000,160
52866 :000,016,004,000,032,003,105
52872 :255,192,013,000,176,040,052
52878 :129,012,064,066,002,112,015
52884 :000,014,124,000,062,127,219
52890 :255,254,127,255,254,127,146
52896 :255,254,127,255,254,127,152
52902 :255,254,063,255,252,015,236
52908 :255,240,003,255,192,000,009
52914 :000,000,000,000,000,000,170
52920 :000,000,000,000,004,000,100
52926 :000,000,000,000,005,000,201
52932 :000,004,120,000,004,064,140
52938 :000,004,000,000,004,000,210
52944 :000,004,000,000,004,000,216
52950 :000,004,000,000,004,000,222
52956 :000,004,000,001,220,000,197
52962 :003,244,000,007,252,000,220
52968 :007,244,000,003,240,000,210
52974 :001,224,000,000,000,000,207
52980 :000,000,000,000,000,000,244
52986 :002,013,013,013,013,013,061

```



"Atari Animator" lets you create cartoons in 128 different colors.

```

M19 DATA 154,141,71,114,10
4,104,10,170,109,139
M20 DATA 114,133,203,232,1
89,139,114,133,204,104
M21 DATA 141,242,6,40,20,1
04,24,109,244,6
M22 DATA 141,244,6,173,246
,6,109,242,6,141
M23 DATA 246,6,24,144,31,7
3,255,24,105,1
M24 DATA 141,242,6,104,141
,245,6,173,246,6
M25 DATA 56,237,245,6,141,
244,6,173,246,6
M26 DATA 237,242,6,141,246
,6,162,3,160,7
M27 DATA 177,203,153,170,9
4,136,16,240,160,15
M28 DATA 177,203,153,42,95
,136,192,8,176,246
M29 DATA 160,23,177,203,15
3,170,95,136,192,16
M30 DATA 176,246,165,203,2
4,105,24,133,203,165
M31 DATA 204,105,0,133,204
,173,49,114,24,105
M32 DATA 0,141,49,114,173,
59,114,105,8,141
M33 DATA 59,114,173,71,114
,24,105,0,141,71
M34 DATA 114,202,200,100,1
73,246,6,141,1,200
M35 DATA 24,105,0,141,2,20
8,24,105,0,141
M36 DATA 3,200,96,8,72,0,7
2,00,72,152
M37 DATA 72,224,72,40,73,1
12,73,104,73,0
M38 DATA 00,00,00,152,0,0,2
24,00,40,0,112
M39 DATA 01,154,01,8,00,00
,00,152,00,224
M40 DATA 00,40,09,112,09,1
84,09

```

## Program 4: Atari Animator, Part 2

```

M10 PDK 559,0:GOSUB 1000:
GOSUB 730:GOSUB 940:0R
APHIC8 0:PDKE 752,1:0D
SUS 200:A=USR(1535)
M12 GOSUB 600:GOSUB 460:0D
TO 20
M30 R=0:TRAP 120:GOSUB 165
0:OPEN #1,6,0,"D18.S"
M40 INPUT #1,SDX:PRINT SD
N:GOTO 40
M50 TRAP 120:X=NUMBER(1):G
OSUB 160:FA=SA:K=NUMBE
R(4):1:GOSUB 140
M60 FAH=INT(FA/256):FAL=FA
-FAH*256:SAH=INT(SA/25
6):BAL=SA-BAH*256
M70 GOSUB 170:IF FN0="" TH
EN 110

```

```

F75 IF DEVE="C": THEN PRIN
T "(DDNN)PRESS RECORD
AND PLAY DN TAPE"
M00 OPEN #1,0,0,FN0:PUT #1
,FAL:PUT #1,FAH:PUT #1
,BAL:PUT #1,SAH
M90 FOR A=F TO SA
M100 PUT #1,PEEK(A):NEXT A
:CLOSE #1
M110 GOSUB 1670:RETURN
M120 TRAP 32767:IF PEEK(19
5)<136 THEN PRINT "
(3 DDNN)SYSTEM ERRDR
"PEEK(195)
M130 PRINT "(DDNN)HIT ANY
KEY TO CONTINUE":POKE
764,255
M140 IF PEEK(764)=255 THEN
140
M150 GOSUB 1670:CLOSE #1:R
ETURN
M160 Y=X-1:BA=(X-(INT(Y/7)
87)>172+(INT(Y/7))>20
40+10360):RETURN
M170 DN0=""FN0=""GOSUB 1
650:PRINT "(4 DDNN)EN
TER FILENAME"
M180 INPUT DN0:IF DN0="" T
HEN 240
M190 PRINT "(3 DDNN)TAPE D
R DISK":POKE 764,255
M200 IF PEEK(764)=45 THEN
DEVE="C":GOTO 230
M210 IF PEEK(764)=50 THEN
DEVE="0":GOTO 230
M220 GOTO 200
M230 FN0=(3,LEN(DN0)+3)=DN0
:FN0(1,2)=DEVE
M240 RETURN
M250 TRAP 120:GOSUB 170:IF
FN0="" THEN 110
M255 IF DEVE="C": THEN PRI
NT "(DDNN)PRESS PLAY
DN TAPE"
M260 DN0=X-4,0,FN0:GET #
1,FAL:GET #1,FAH:GET
#1,BAL:GET #1,SAH:FA=
FAL+256*FAH:SA=FA+25
6*SAH
M270 FDR A=F TO SA+64:GET
#1,0:POKE A,0:NEXT A
GOSUB 1670:CLOSE #1:
RETURN
M280 SDX=""(3 N)(DDNN)
(4 LEFT)(0)
(3 SPACES)(V)(DDNN)
(5 LEFT)(0)
(3 SPACES)(V)(DDNN)
(5 LEFT)(0)
(3 SPACES)(V)(DDNN)
(4 LEFT)(3 M)(4 UP)
(RIGHT)"
M290 PRINT "(CLEAR)"
M300 FOR A=0 TO 12 STEP 6:
POSITION 2,A
M310 FOR S=1 TO 7:PRINT BD
X:PRINT 0
M320 NEXT A
M330 POSITION 14,5:PRINT "1
(4 SPACES)2
(4 SPACES)3
(4 SPACES)4
(4 SPACES)5
(4 SPACES)6
(4 SPACES)7"
M340 POSITION 14,11:PRINT "
6(4 SPACES)9
(4 SPACES)10
(4 SPACES)11
(4 SPACES)12
(4 SPACES)13
(4 SPACES)14"
M350 POSITION 14,17:PRINT "
15(3 SPACES)16

```

## Program 3: Atari Animator, Part 1

Version by Kevin Mykityn, Editorial Programmer

Please refer to "COMPUTER'S Guide to Typing in Programs" before entering these listings.

```

M0 GRAPHICS 0:PDKE 752,1:P
RINT "(CLEAR)"(5 DDNN)
(12 SPACES)PLEASE WAIT":
PRINT "(DDNN)"
(7 SPACES)THE SCREEN W1
LL BLANK
M1 CH=0:FDR A=20000 TO 293
66:READ #1:PDKE A,B1CH=C
H00:NEXT A:IF CH<43015
THEN PRINT "ERRDR IN D
ATA":END
M2 RUN "D1PART2"
M3 DATA 169,3,141,242,6,16
9,0,141,243,6
M5 DATA 165,00,133,205,165
,09,133,206,169,0
M6 DATA 141,241,6,169,0,14
1,240,6,160,7
M7 DATA 56,177,205,201,12,
240,1,24,110,240
M8 DATA 6,136,16,242,173,2
40,6,174,241,6
M9 DATA 157,0,96,32,190,11
3,173,241,6,24
M10 DATA 105,0,141,241,6,2
01,90,176,47,206
M11 DATA 242,6,205,39,169,
3,141,242,6,24
M12 DATA 190,113,32,190,11
3,173,241,6,56,233
M13 DATA 23,141,241,6,206,
243,6,200,14,169
M14 DATA 0,141,243,6,173,2
41,6,24,105,16
M15 DATA 141,241,6,24,144,
163,104,96,165,200
M16 DATA 24,105,0,133,205,
165,206,105,0,133
M17 DATA 206,96,163,20,197
,20,240,252,104,169
M18 DATA 170,149,149,114,16
9,34,141,59,114,169

```

```

(3 SPACES)17
(3 SPACES)18
(3 SPACES)19
(3 SPACES)20
(3 SPACES)21*
N368 E=-9;FOR S=0 TO 30 ST
EP 5:E=E+Y
N370 FOR A=1 TO 13 STEP 6:
CHAR=33
N380 FOR O=0 TO 2:POSITION
3+B,A+O
N390 FOR C=0 TO 2:PRINT CH
R:(CHAR+C+E);
N400 NEXT C:PRINT "(DOWN)
(3 LEFT)";:CHAR=CHAR+
3
N410 NEXT O:NEXT A:NEXT B
N420 POSITION 1,19:PRINT "
START PICTURE
(7 SPACES)END PICTURE
"
N430 POSITION 1,20:PRINT "
PICTURE NUMBER
(6 SPACES)OVERALL SPE
ED"
N440 POSITION 1,21:PRINT "
SWITCHING SPEED "
N450 RETURN
N460 JOY=STICK(0)
N470 ON JOY-4 GOTO 560,560
,480,560,560,560,500,
560,520,540,560
N480 IF NUMBER(ARRPOS)=21
THEN NUMBER(ARRPOS)=0
N490 NUMBER(ARRPOS)=NUMBER
(ARRPOS)+1:GOTO 560
N500 IF NUMBER(ARRPOS)=1 T
HEN NUMBER(ARRPOS)=22
N510 NUMBER(ARRPOS)=NUMBER
(ARRPOS)-1:GOTO 560
N520 GOSUB 660:IF ARRPOS=5
THEN ARRPOS=0
N530 ARRPOS=ARRPOS+1:GOTO
560
N540 GOSUB 660:IF ARRPOS=1
THEN ARRPOS=6
N550 ARRPOS=ARRPOS-1
N560 POSITION COORD(ARRPOS
,1),COORD(ARRPOS,2)
N570 PRINT ">"
N580 IF STRIG(0)=0 THEN GO
SUB 1540
N590 KEY=PEEK(764):POKE 76
4,255
N600 IF KEY=42 THEN GOSUB
1100
N610 IF KEY=62 THEN GOSUB
50
N620 IF KEY=0 THEN GOSUB 2
50
N630 IF KEY=58 THEN GOSUB
30
N640 IF KEY=18 THEN FOR A=
0 TO 511:POKE A+CHAR1
,0:POKE A+CHAR2,0:POK
E A+CHAR3,0:NEXT A
N650 RETURN
N660 POSITION COORD(ARRPOS
,1),COORD(ARRPOS,2)
N670 PRINT " *RETURN
N680 FOR A=1 TO 4
N690 POSITION COORD(A,1)+1
6+(A*4),COORD(A,2)
N700 POSITION NUMBER(A); "
N710 NEXT A:POSITION 36,20
:PRINT NUMBER(A)-10;"
"
N720 RETURN
N730 RESTORE 760:A=1535
N740 READ B:IF B=256 THEN
A=UBR(1535):RETURN
N750 POKE A,B:A=A+1:GOTO 7
40
N760 DATA 104,173,48,2,141
,203,0
N770 DATA 173,49,2,141,204
,0
N780 DATA 160,28,169,2,145
,203
N790 DATA 136,192,5,200,24
,9,162
N800 DATA 5,189,102,6,160,
177
N810 DATA 203,9,120,145,20
3,202
N820 DATA 16,243,169,64,14
,1,0
N830 DATA 2,169,6,141,1,2
N840 DATA 169,255,141,100,
6,173
N850 DATA 11,212,200,251,1
69,192
N860 DATA 141,14,212,96,72
,130
N870 DATA 72,230,100,6,173
,100
N880 DATA 6,201,6,200,5,16
,9
N890 DATA 0,141,100,6,170,
189
N900 DATA 96,6,141,10,212,
141
N910 DATA 9,212,104,170,10
4,64
N920 DATA 72,224,00,224,00
,224
N930 DATA 3,0,11,14,17,20,
256
N940 ORIGINAL=57344:CHAR1=
10432:CHAR2=20480:CHA
R3=25520:CHAR0=24768
N950 FOR A=ORIGINAL TO ORI
GINAL+1024
N960 Q=ORIGINAL:V=PEEK(A
)
N970 POKE CHAR1+0,V:POKE C
HAR2+0,V:POKE CHAR3+0
,V:POKE CHAR0+0,V:NEX
T A
N980 RESTORE 990:FOR A=246
64 TO 24687:READ B:PO
KE A,B:NEXT A:RETURN
N990 DATA 48,48,48,255,255
,48,48,48,0,126,126,1
,26,126,126,126,0,255,
129,129,129,129,129,1
29,255
N1000 DIM BOX$(50):DIM NUM
BER$(5):DIM B$(5):DIM
BINARY$(64):DIM FNS
(15):DIM DEV$(2):DIM
ONS(17):DIM COORD(5
,2)
N1010 POKE 752,1:RESTORE 1
030:FOR A=1 TO 51:NUM
BER(A)=1:FOR B=1 TO
2
N1020 READ C:COORD(A,B)=C:
NEXT B:NEXT A
N1030 DATA 0,19,0,20,0,21,
20,19,20,20
N1040 ARRPOS=RESTORE 106
0:FOR A=1 TO 64:READ
B
N1050 BINARY$(A)=CHR$(B
I+44):NEXT A
N1060 DATA 1,1,1,1,1,1,1,0
,1,1,0,1,1,1,0,0,1,0
,1,1,1,0,1,0,1,0,0,1
,1,0,0,0,0,1,1,1,0,1
,1,0,0,1,0,1,0,1,0,1
,0,0,0,1,1,0,0,0,0,0
N1070 DATA 0,0,1,1,0,0,1,0
,0,0,0,1,0,0,0,0
N1080 OLI=PEEK(560)+256:PE
EK(561):ODL=PEEK(OLI+
4):OOH=PEEK(OLI+5)
N1090 RETURN
N1100 GOSUB 1110:GOTO 1200
N1110 NU=NUMBER(2)-1:POKE
756,96
N1120 BA=(NUMBER(2)-(INT(N
U/7)+87)):B72+(INT(NU/
7)):B2048+10360
N1130 GOSUB 1650
N1140 R=0:B=0:NR=B:FOR A=B
A TO BA+71:P=PEEK(A)
H$=INT(P/16)+1:L$=P-
16*(H$-1)+1
N1150 POSITION 0,0:PRINT
BINARY$(H$+4-3,H$+
4):BINARY$(L$+4-3,L$+
4)
N1170 R=R+1:IF R=NR THEN R
=NR:B=B+1:IF B=3 T
HEN R=R+B:NR=NR+B:B=
0
N1180 NEXT A
N1190 RETURN
N1200 POSITION 31,16:PRINT
NUMBER(2)
N1210 FOR A=0 TO 2:FOR B=0
TO 2
N1220 POSITION 36+B,11+A:P
RINT CHR$(33+B+A*3)
N1230 NEXT B:NEXT A
N1240 A=UBR(29000):IF FL=0
THEN LOCATE 0X,0Y,1
N1250 OX=0X:OY=0Y:POSITION
0X,OY:PRINT "*"
N1260 JOY=STICK(0)-4
N1270 ON JOY GOTO 1200,120
0,1300,1360,1310,132
0,1330,1360,1340,135
0,1360
N1280 OX=OX+1:OY=OY+1:GOTO
1360
N1290 OX=OX+1:OY=OY-1:GOTO
1360
N1300 OX=OX+1:GOTO 1360
N1310 OX=OX-1:OY=OY+1:GOTO
1360
N1320 OX=OX-1:OY=OY-1:GOTO
1360
N1330 OX=OX-1:GOTO 1360
N1340 OY=OY+1:GOTO 1360
N1350 OY=OY-1:GOTO 1360
N1360 IF OX>23 THEN OX=0
N1370 IF OY>23 THEN OY=0
N1380 IF OX<0 THEN OX=23
N1390 IF OY<0 THEN OY=23
N1400 IF STRIG(0)=0 AND FL
=0 THEN FL=1:Z=(Z+44
)+1+44
N1410 IF STRIG(0)<>0 THEN
FL=0
N1420 POSITION 0X,OY:PRINT
CHR$(Z)
N1430 KEY=PEEK(764):POKE 7
64,255
N1440 IF KEY=18 THEN FOR A
=0 TO 23:FOR B=0 TO
23:POSITION A,B:PRIN
T CHR$(45):NEXT B:N
EXT A
N1450 IF KEY=42 THEN 1510
N1460 IF KEY=6 AND NUMBER(
2)<21 THEN GOSUB 152
0:GOTO 1100
N1470 IF KEY=14 AND NUMBER
(2)>1 THEN GOSUB 152
0:NUMBER(2)=NUMBER(2
)-1:GOTO 1100
N1480 IF KEY=61 THEN SHP=N
UMBER(2)
N1490 IF KEY=10 THEN TEMP=
NUMBER(2):TEMP2=SA:N
UMBER(2)=SHP:GOSUB 1
110:NUMBER(2)=TEMP:0

```

```

A=TEMP2:GOTO 1200
N 1500 GOTO 1240
U 1510 GOSUB 1670:GOSUB 152
0:RETURN
N 1520 FOR A=0 TO 71:POKE 8
A,A,PEEK(24584+A):NEXT
A
N 1530 RETURN
N 1540 IF STRIG(0)=0 THEN 1
540
N 1550 GOSUB 1650:POKE 559,
46:POKE 623,1:POKE 7
05,8:POKE 706,8:POKE
707,8:POKE 53277,3:
POKE 54279,92
N 1560 SP=NUMBER(1):EP=NUM
BER(4):DIR=SGN(EP-NB)
:SB=NUMBER(3):OS=NUM
BER(5)-10
N 1570 S=ABS(OS)/3*255:IF 0
<0 THEN Q=255-INT(S
/255):R=S-(255-Q)*25
5:G=Q*256+R
N 1580 OS=S
U 1590 FOR PN=BP TO EP STEP
DIR
N 1600 A=USR(29132,PN,OS)
N 1610 TD=TD-1:IF TD<0 THEN
TD=22-SB:NEXT PN:GO
TO 1590
N 1620 KEY=PEEK(764):POKE 7
05,KEY:POKE 706,KEY:
POKE 707,KEY
N 1630 IF STRIG(0)<0 THEN
1600
N 1640 FOR A=53248 TO 53251
:POKE A,220:NEXT A:IB
OSUB 1670:RETURN
N 1650 POKE 54286,64:POKE 8
9,8:POKE 89,100:POKE
DLI+4,0:POKE DLI+5,DD
H:POKE 89,DD:POKE 8
9,DDH:RETURN
N 1660 T=PEEK(106):POKE 106,
104:PRINT CHR$(125)
:POKE 106,T:RETURN
N 1670 A=USR(1535):POKE DLI
+4,ODL:POKE DLI+5,DD
H:POKE 89,DDL:POKE 8
9,DDH:RETURN

```

## Program 5: Apple ANIMATOR2 Maker

Version by Tim Victor, Editorial  
Programmer

Please refer to "COMPUTE's Guide to Typing  
In Programs" before entering these listings

```

N 800 FOR I = 126 TO 256 TO I + 1
455: READ A: POKE I, A: NEX
T
N 90 PRINT CHR$(4) : "BSAVE ANIM
ATOR2, A$TWO, L$DS00": END
N 100 DATA 0,0,0,0,0,0,0,0,0,0,
0
N 110 DATA 0,0,0,0,0,0,216,120,13
3,69,134,70
N 120 DATA 132,71,166,7,10,10,1,
70,4,16,62,40
N 130 DATA 4,1,1,232,232,10,13
4,27,24,101,4
N 140 DATA 133,26,144,2,230,27,
165,40,133,8,165
N 150 DATA 41,41,3,5,230,133,9,
162,8,160,0
N 160 DATA 177,26,36,50,40,2,73
,127,164,36,145
N 170 DATA 0,230,26,200,2,230,2
7,165,9,24,105
N 180 DATA 4,133,9,202,200,226,
165,69,166,70,164
N 190 DATA 71,80,76,240,253,120
,120,120,120,190,120

```

```

55 200 DATA 120,120,120,120,120,
190,120,190,120,120,120
N 210 DATA 100,230,176,152,120,
152,120,120,100,230,246
N 220 DATA 230,230,100,120,120,
152,156,152,152,152,100
N 230 DATA 120,120,100,230,176,
140,230,254,120,120,100
N 240 DATA 230,176,224,230,100,
120,120,176,104,100,254
N 250 DATA 176,176,120,120,254,
134,190,224,230,100,120
N 260 DATA 120,100,134,190,230,
230,100,120,120,254,224
N 270 DATA 176,152,140,140,120,
120,100,230,100,230,230
N 280 DATA 100,120,100,100,230,
230,252,176,152,120,120
N 290 DATA 152,176,254,254,176,
152,120,120,190,190,190
N 300 DATA 190,190,190,120,0,0,
0,0,0,0,0
N 310 DATA 0,0,0,0,0,0,0,0,0,0,
0
N 320 DATA 0,0,0,0,0,0,0,0,0,0,
0
N 330 DATA 0,0,0,120,120,152,10
0,100,152,120,120
N 340 DATA 120,252,230,230,254,
230,230,120,120,190,230
N 350 DATA 230,190,230,254,120,
120,100,230,134,134,230
N 360 DATA 190,120,120,190,230,
230,230,230,190,120,120
N 370 DATA 254,134,134,190,134,
254,120,120,254,134,134
N 380 DATA 190,134,134,120,120,
100,230,134,246,230,190
N 390 DATA 120,120,230,230,230,
254,230,120,120,120,152
N 400 DATA 152,152,152,152,152,
120,120,224,224,224,224
N 410 DATA 230,100,120,120,230,
230,102,100,230,230,120
N 420 DATA 120,134,134,134,134,
134,254,120,120,230,254
N 430 DATA 230,230,230,230,120,
120,190,230,230,230,230
N 440 DATA 230,120,120,100,230,
230,230,230,100,120,120
N 450 DATA 190,230,230,190,134,
134,120,120,100,230,230
N 460 DATA 230,102,230,120,120,
190,230,230,190,230,230
N 470 DATA 120,120,100,230,140,
176,230,190,120,120,254
N 480 DATA 152,152,152,152,152,
120,120,230,230,230,230
N 490 DATA 230,190,120,120,230,
230,230,230,230,152,120
N 500 DATA 120,230,230,230,230,
254,230,120,120,230,230
N 510 DATA 230,100,230,230,152,
120,230,230,230,100,120
N 520 DATA 152,120,120,254,176,
152,140,134,254,120,0
N 530 DATA 0,0,0,0,0,0,0,0,0,0,
162,7
N 540 DATA 169,196,133,254,169,
132,133,255,177,254,73
N 550 DATA 127,145,250,200,200,
247,230,255,202,200,242
N 560 DATA 90,0,0,0,0,0,76,21,1
28,76,85
N 570 DATA 120,76,130,120,76,23
9,120,76,6,130,76
N 580 DATA 160,129,76,210,129,1,
69,2,141,100,131,169
N 590 DATA 24,141,101,131,32,31,
130,176,48,32,71
N 600 DATA 131,176,43,32,156,13
1,176,30,173,170,131
N 610 DATA 133,252,173,179,131,
133,253,32,223,130,32

```

```

N 620 DATA 95,130,32,160,130,23
0,104,131,165,252,24
N 630 DATA 100,3,133,252,144,2,
230,253,200,101,131
N 640 DATA 200,220,96,169,2,141
,100,131,169,24,141
N 650 DATA 131,131,32,71,131,17
6,29,32,156,131,76
N 660 DATA 24,160,3,169,0,153,1
96,131,136,16,250
N 670 DATA 32,223,130,32,160,13
0,230,104,131,200,101
N 680 DATA 131,200,242,96,165,5,
6,164,57,201,100,200
N 690 DATA 4,192,120,240,14,141
,190,131,140,191,131
N 700 DATA 169,106,133,56,160,1
20,132,57,32,121,129
N 710 DATA 169,0,141,194,131,32
,31,130,176,13,173
N 720 DATA 170,131,141,197,120,
173,179,131,141,190,120
N 730 DATA 96,32,6,130,0,145,40
,169,40,141,0
N 740 DATA 2,141,1,2,173,255,25
5,201,100,144,7
N 750 DATA 233,100,230,0,2,200,
245,201,10,144,7
N 760 DATA 233,10,230,1,2,200,2
45,100,40,141,2
N 770 DATA 2,162,3,169,141,230,
197,120,200,3,230
N 780 DATA 190,120,96,32,121,12
9,169,255,141,194,131
N 790 DATA 32,31,130,176,23,173
,170,131,141,0,141,129
N 800 DATA 173,179,131,141,85,1
29,169,196,141,110,129
N 810 DATA 169,131,141,111,129,
96,32,6,130,0,44
N 820 DATA 191,131,40,3,76,240,
253,201,176,144,4
N 830 DATA 201,106,144,70,72,14
0,195,131,173,110
N 840 DATA 129,233,160,240,53,1
41,102,131,169,0,160
N 850 DATA 201,26,176,214,10,14
1,60,129,10,10,105
N 860 DATA 255,176,204,121,196,
131,56,233,176,200,204
N 870 DATA 102,131,200,230,141,
255,255,230,0,129,200
N 880 DATA 3,230,85,129,169,196
,141,110,129,169,131
N 890 DATA 141,111,129,172,195,
131,104,96,141,255,255
N 900 DATA 230,110,129,200,3,23
0,111,129,96,165,50
N 910 DATA 164,55,201,23,200,4,
192,129,240,14,141
N 920 DATA 192,131,140,193,131,
169,23,133,54,160,129
N 930 DATA 132,50,96,32,31,130,
176,56,173,170,131
N 940 DATA 131,130,129,200,3,20
0,179,131,200,190,129
N 950 DATA 173,179,131,141,204,
129,169,227,133,252,160
N 960 DATA 142,133,253,160,0,17
7,252,160,72,145,252
N 970 DATA 165,252,200,2,190,25
3,190,252,169,255,197
N 980 DATA 252,200,234,169,255,
197,253,200,220,96,32
N 990 DATA 31,130,176,46,173,17
0,131,133,252,173,179
N 1000 DATA 131,133,253,169,220
,141,250,120,169,142,141
N 1010 DATA 0,130,160,72,177,25
2,160,0,145,252,230
N 1020 DATA 252,200,2,230,253,1
69,255,197,252,200,236
N 1030 DATA 169,255,197,253,200
,230,96,173,190,131,172

```

```

R 1040 DATA 191,131,248,4,133,5
      6,132,57,173,192,131
R 1050 DATA 172,193,131,248,4,1
      33,54,132,55,96,169
R 1060 DATA 196,141,178,131,169
      132,141,179,131,32,165
R 1070 DATA 131,281,21,144,1,96
      141,178,131,169,0
R 1080 DATA 141,177,131,168,3,3
      2,64,138,168,3,173
R 1090 DATA 178,151,18,46,177,1
      31,136,288,249,141,176
R 1100 DATA 131,24,189,178,131,
      141,178,131,173,179,131
R 1110 DATA 189,177,131,141,179
      131,24,96,172,180,131
R 1120 DATA 208,148,182,131,169
      0,153,196,131,136,177
R 1130 DATA 252,153,196,131,136
      16,248,173,196,131,9
R 1140 DATA 127,141,183,131,172
      187,131,248,21,162,0
R 1150 DATA 14,196,131,189,196,
      131,18,62,197,131,232
R 1160 DATA 236,182,131,208,243
      136,280,235,172,182,131
R 1170 DATA 185,196,131,9,128,4
      5,183,131,153,196,131
R 1180 DATA 136,16,242,96,172,1
      87,131,185,289,136,172
R 1190 DATA 182,131,49,254,25,1
      96,131,45,254,136,185
R 1200 DATA 196,131,145,254,136
      288,248,172,187,131,15
R 1210 DATA 216,138,168,4,49,25
      4,13,196,131,145,254
R 1220 DATA 96,127,128,124,128,
      112,96,64,0,1,3
R 1230 DATA 7,15,31,63,173,184,
      131,41,63,168,185
R 1240 DATA 7,131,5,258,133,255
      173,184,131,41,8
R 1250 DATA 248,2,169,128,24,44
      184,131,112,4,16
R 1260 DATA 4,185,48,185,48,189
      186,131,133,254,96
R 1270 DATA 6,4,8,12,16,28,24,2
      8,0,4,0
R 1280 DATA 12,16,28,24,28,1,5,
      9,13,17,21
R 1290 DATA 25,29,1,5,9,13,17,2
      1,25,29,2
R 1300 DATA 6,18,14,18,22,26,38
      2,6,18,14
R 1310 DATA 18,22,26,38,3,7,11,
      15,19,23,27
R 1320 DATA 31,3,7,11,15,19,23,
      27,31,169,0
R 1330 DATA 141,186,131,141,187
      131,32,165,131,141,185
R 1340 DATA 131,192,1,144,18,24
      0,1,96,281,24,144
R 1350 DATA 1,96,169,36,141,186
      131,169,4,141,187
R 1360 DATA 131,169,0,141,188,1
      31,169,224,141,189,131
R 1370 DATA 173,185,131,285,189
      131,144,4,237,189,131
R 1380 DATA 56,48,188,131,78,18
      9,131,144,239,24,189
R 1390 DATA 187,131,141,187,131
      24,173,188,131,189,186
R 1400 DATA 131,141,186,131,24,
      96,32,165,131,141,184
R 1410 DATA 131,281,192,96,32,1
      77,0,32,225,165
R 1420 DATA 161,164,168,96

```

### Program 6: Apple Animator

```

R 100 DIM EX(2),PS(8),CHS(7)
      ,HMS(14):O# = CHR$(4):P
      RINT D#:"BLDAD ANIMATOR2"

```



In this demonstration of "Apple Animator," a motorcyclist performs a daring wheelie stand.

```

R 110 FOR I = 0 TO 8: READ MS(I)
      ): NEXT I: FOR I = 0 TO 7:
      READ CHS(I): NEXT I: FOR
      I = 0 TO 14: READ HMS(I):
      NEXT I
R 120 GOSUB 530
R 130 A# = "": FOR I = 1 TO 72:
      A# = A# + "0":A = FRE (0)
      ): NEXT I
R 140 GOSUB 650: GOSUB 590
R 150 F = 1:A# = 1:AE = 28:A# =
      0:AF = 10
R 160 ONERR GOTO 1390
R 170 GOSUB 710: IF MQ = 1 THEN
      HOME: TEXT: END
R 180 C = PEEK (49152): IF C <
      128 THEN 180
R 190 POKE 49168,0: IF C = 136
      THEN F = F - 1 + 20:8 (F
      = 1): GOTO 170
R 200 IF C = 149 THEN F = F + 1
      - 20:8 (F = 20): GOTO 17
      0
R 210 C = C - 175: IF C < 1 OR
      C > 22 THEN 180
R 220 IF C < 11 THEN 250
R 230 IF C < 18 THEN 250
R 240 C = C - 7
R 250 MQ = 0: ON C GOSUB 260,11
      28,1210,1500,1510,1520,16
      68,1670,1680,1690,1370,12
      50,1470,1480,1490:T = FRE
      (0): GOTO 170
R 260 GOSUB 560: GOSUB 780: VTA
      B 19: HTAB 14: PRINT "EDI
      TING BOX":J#A
R 270 PRINT "PRESS ESC TO CANCE
      L": HTAB 7: PRINT "RETURN
      FOR SAME": PRINT "PRINT
      "STORE RESULT IN BOX":J:
      R# = "
R 280 XC = 21 + LEN (R#): VTAB
      23: HTAB XC: PRINT "J:
R 290 C = PEEK (49152): IF C <
      128 THEN 290
R 300 POKE 49168,0: IF C = 141
      OR C = 155 THEN VTAB 23:
      HTAB XC: PRINT "J: GOTO
      360
R 310 IF C < > 136 AND C < > 25
      5 THEN 340
R 320 VTAB 23: HTAB XC: PRINT =
      "J: IF LEN (R#) < 2 THEN
      R# = "": GOTO 280
R 330 R# = LEFT$ (R#, LEN (R#)
      - 1): GOTO 280
R 340 IF C < 176 OR C > 185 THE
      N 290
R 350 VTAB 23: HTAB XC: PRINT C
      HR# (C - 128):R# = R# +
      CHR# (C - 128):O = FRE (0)
      ): GOTO 280
R 360 IF C = 155 THEN GOSUB 660
      : RETURN
R 370 IF R# = "" THEN AA = A: B

```

```

OTO 390
R 380 AA = VAL (R#): IF AA > 20
      THEN GOSUB 660: RETURN
R 390 H#R2: HOME: GOSUB 1870
R 400 X# = 184:Y# = 44:O# = 65:
      OY = 80: GOSUB 980
R 410 VTAB 7: FOR Q = 0 TO 8: H
      TAB 28: INVERSE: PRINT M
      IO# ("123456789"):Q = 1,1)
      ): NORMAL: PRINT "J:H#(
      Q): NEXT
R 420 CALL 32768,A,208,12:XC =
      0:YC = 0:Q# = 0: GOSUB 87
      0
R 430 IF OF THEN GOSUB 590: RET
      URN
R 440 SC = INT (YC / 8):SA = 10
      24 + YC % 128 - SC % 984
      + XC:CCS = "
R 450 CC# = CCS:CC# = CHR# ( PE
      CE (SA) - 128): HTAB XC +
      1: VTAB YC + 1: PRINT OC
      S:
R 460 C = PEEK (49152): IF C <
      128 THEN T = FRE (0): GOTO
      0 450
R 470 POKE 49168,0: IF CC# = "
      THEN HTAB XC + 1: VTAB
      YC + 1: PRINT CC#
R 480 FOR Q = 1 TO 13: IF C < >
      ASC ("12345678901234567
      89"):Q,1) + 128 THEN NEXT
R 490 ON C GOSUB 750,760,770,78
      0,840,850,870,880,890,990
      1020,1050,1060: GOTO 430
R 500 I = A % 28 - 24 - 200:8 (
      A > 10):J = 26 + 80:8 (A
      > 10): CALL 32768,A,1,J
R 510 X = FRE (0): RETURN
R 520 FOR I = 1 TO 20: GOSUB 50
      0: NEXT I: RETURN
R 530 POKE 6,0: POKE 7,126: IF
      PEEK (40640) = 76 THEN 55
      0
R 540 POKE 54,16: POKE 55,126:
      CALL 1002: RETURN
R 550 PRINT: PRINT CHR# (4):"P
      R#A#7E10": RETURN
R 560 A = F: RETURN
R 570 GOSUB 580: HTAB 11: INPUT
      "WHICH BOX?":A: A = INT (A
      ): IF A < 1 OR A > 20 THE
      N 570
R 580 VTAB 22: HTAB 11: PRINT SP
      C (39): RETURN
R 590 HOME: H#R2: HCOLOR = 3
R 600 FOR J = 25 TO 185 STEP 80
      FOR I = 3 TO 1 + 9 * 28
      STEP 28: FOR P = 0 TO 1 +
      22 + P,J - P TO 1 + 22 +
      P,J - 25 + P TO 1 + P,J +
      25 + P TO 1 + P,J - P
R 620 NEXT I: NEXT J: NEXT
R 630 FOR J = 0 TO 1: FOR I = 1
      TO 181: HTAB 11: # 4 - 2: V
      TAB J % 10 + 3: PRINT I +
      J % 10: NEXT I: NEXT
R 640 GOSUB 520: GOTO 660
R 650 POKE 242,0: CALL 32777,0:
      FOR I = 0 TO 20: PRINT A
      #: NEXT I: CALL 32780: RET
      URN
R 660 GOSUB 780:OY = 2:Y# = 140
      :O# = 275:O# = 48: GOSUB
      980
R 670 VTAB 19: FOR I = 0 TO 4:
      HTAB 21: INVERSE: PRINT I
      ): NORMAL: PRINT "J:H#(
      I): HTAB 17: INVERSE:
      PRINT I + 5:
R 680 NORMAL: PRINT "J:H#(I
      + 5): HTAB 32: INVERSE:
      PRINT CHR# (65 + I): NO

```

```

RMAL: PRINT " "MM(I +
10): NEXT
2 690 RETURN
3 700 HTAB 1: VTAB 18: FDR 0 =
1 TD 7: PRINT SPC( 40): N
EXT: RETURN
4 710 VTAB 1: HTAB 1: PRINT "FR
AME "F": "
5 720 HTAB 10: PRINT "RANGE "A
B": " : HTAB 18: PRINT "
":AE": "
6 730 HTAB 22: PRINT "SPEED "A
S": " : HTAB 32: PRINT "
PAUSE "AP": "
7 740 RETURN
8 750 XC = XC - (XC > 0): GOTO
790
9 760 XC = XC + (XC < 20): GOTO
790
10 770 YC = YC - (YC > 0): GOTO
790
11 780 YC = YC + (YC < 23): GOTO
790
12 790 XB = INT (X / 7): PM = 2
- (XC - 7 * XB): T = INT (
EX(XB, YC) / PM): ON OF GO
TO 800,820: RETURN
13 800 IF T = 2 * INT (T / 2) TH
EN EX(XB, YC) = EX(XB, YC
+ PM): HTAB XC + 1: VTAB Y
C + 1: PRINT "I":
14 810 RETURN
15 820 IF T < 2 * INT (T / 2)
THEN EX(XB, YC) = EX(XB, YC
- PM): HTAB XC + 1: VTAB
YC + 1: PRINT "I":
16 830 RETURN
17 840 GF = 1: RETURN
18 850 OF = 1: GOSUB 860: HTAB 2
9: VTAB 8: PRINT "S": GO
TO 790
19 860 VTAB 8: FOR I = 1 TO 3: H
TAB 29: PRINT " ": NEXT I
RETURN
20 870 DF = 0: GOSUB 860: HTAB 2
9: VTAB 9: PRINT "S": RE
TURN
21 880 DF = 2: GOSUB 860: HTAB 2
9: VTAB 10: PRINT "S": G
OTO 790
22 890 XP = 149: YP = 140: DX = 12
BI: OY = 40: GDSUB 980
23 900 FDR I = 0 TO 23: VTAB 19 +
I: FDR J = 0 TO 21: HTAB
23 + J * J: INVERSE: PRI
NT I + J * J: NORMAL: P
RINT " " : CH$(I + J * J):
NEXT J: NEXT I
24 910 C = PEEK (49152): IF C <
128 THEN 910
25 920 VTAB 18: FDR I = 0 TO 5:
HTAB 22: PRINT SPC( 19):
NEXT: POKE 49168,0: C = C
- 176
26 930 IF C < 0 DR C > 7 THEN RE
TURN
27 940 T = C - 4 * INT (C / 4): P
0 = 42 * T + (T / 1): P1 =
P0: IF P0 = 42 DR P0 = 0
5 THEN P1 = 127 - P0
28 950 IF C > 3 THEN P0 = P0 + 1
29: P1 = P1 + 128
29 960 T = T + (T / 2): FDR I =
0 TO 23: VTAB I + 1: HTAB
1:EX(0,1) = P0:EX(1,1) =
P1:EX(2,1) = P0
30 970 FDR J = 1 TO 10: PRINT MI
D$( "1,1,1,1",T + 1,2): N
EXT: PRINT MID$( "1,1,1,1",T
+ 1,1): NEXT J: RET
URN
31 980 FDR P = 0 TO 1: HPLDT XP
+ P, YP + P TO DX + DX - P
, YP + P TO DX + OX - P, YP
+ OY - P TO XP + P, YP + P:
NEXT: RETURN
32 990 GF = 1: GOSUB 1000: RETUR
N
33 1000 POKE 242,0: CALL 32777,A
A
34 1010 FDR I = 0 TO 23: FDR J =
0 TO 21: PRINT EX(J,1):
NEXT J: NEXT I: CALL 32780
1: RETURN
35 1020 HOME: I FOR I = 0 TO 23:
FDR J = 0 TO 210 = 127 -
EX(J,1): IF 0 < 0 THEN
0 = 0 + 256
36 1130 GOSUB 700: IF A < 20 THE
N CALL 32783,A
37 1140 POKE 242,0: CALL 32777,A
: PRINT A$: CALL 32780:
FOR A = A TO 20: GOSUB 5
00: NEXT
38 1150 GOSUB 660: RETURN
39 1160 PRINT " -REALLY":
40 1170 C = PEEK (49152): IF C <
128 THEN 1170
41 1180 POKE 49168,0: IF C = 206
THEN PRINT "NO": RETURN
42 1190 IF C = 217 THEN PRINT "Y
ES": RETURN
43 1200 GOTO 1170
44 1210 GOSUB 560: GOSUB 700: VT
AB 19: HTAB 10: PRINT "O
ELETE RDX "A: GOSUB 11
60: IF C = 206 THEN 1240
45 1220 GOSUB 700: IF A < 20 THE
N CALL 32786,A
46 1230 POKE 242,0: CALL 32777,2
0: PRINT A$: CALL 32780:
FDR A = A TO 20: GOSUB
500: NEXT
47 1240 GOSUB 660: RETURN
48 1250 T0 = "SAVE": GOSUB 1270:
IF LEN (N0) < 0 THEN
PRINT + T0 "B$AVE
":N0",A0B4C4,L$SEB"
49 1260 GOTO 590
50 1270 HOME: TEXT: VTAB 2: PR
INT "ESC TO CANCEL, RET
RN FOR CATALOG"
51 1280 PRINT: PRINT F0: FILEN
AME":N0 = ""
52 1290 GOSUB 1320: IF C0 = CHR$(
27) THEN N0 = "": RETU
RN
53 1300 IF N0 = "" THEN PRINT:
PRINT D0:"CATALOG": GOTO
1280
54 1310 RETURN
55 1320 T = FRE (0): GET C0: IF
C0 = CHR$( 13) DR C0 = C
HR$( 27) THEN RETURN
56 1330 IF C0 < > CHR$( 127) AND
C0 < > CHR$( 8) THEN N0 =
N0 + C0: PRINT C0: S
DID 1320
57 1340 IF N0 = "" THEN 1320
58 1350 HTAB LEN (N0) + 1: PRIN
T " " : HTAB LEN (N0) +
14: IF LEN (N0) = 1 THEN
N0 = "": GOTO 1320
59 1360 N0 = LEFT$( N0, LEN (N0)
- 1): GOTO 1320
60 1370 F0 = "LOAD": GOSUB 1270:
IF LEN (N0) < 0 THEN
PRINT: PRINT D0:"BLDAD
":N0",A0B4C4"
61 1380 GOTO 590
62 1390 PRINT: PRINT "AN ERRDR
HAS OCCURRED"
63 1400 PRINT "MAKE SURE THAT YO
U HAVE A FORMATTED"
64 1410 PRINT " DISK IN THE DRIV
E"
65 1420 PRINT: PRINT "PRESS ANY

```

```

KEY TO CONTINUE"
66 1430 GET N0: GOSUB 590: GOTO
170
67 1440 GOSUB 700: VTAB 19: HTAB
10: PRINT "QUIT ANIMATO
R": GOSUB 1160: IF C =
217 THEN N0 = 1: RETURN
68 1450 GOSUB 660: RETURN
69 1460 CALL 32736: GOTO 520
70 1470 GOSUB 700: VTAB 19: HTAB
19: PRINT "CLEAR ALL BOX
ES": GOSUB 1160: IF C =
206 THEN 1490
71 1480 GOSUB 700: GOSUB 650: GO
SUB 520
72 1490 GOSUB 660: RETURN
73 1500 AB = F: RETURN
74 1510 AE = F: RETURN
75 1520 A = AB: OF = 0: AR = 0: AX
= 0
76 1530 CALL 32771,AD,60: CALL 3
2768,AX,AX,60:AD = AX: IF
OF = 1 THEN RETURN
77 1540 C = PEEK (49152): IF C >
128 THEN POKE 49168,0: C
GOSUB 1600
78 1550 FDR I = 0 TO AP * 5: NEX
T I: AR = AR + AS: IF AR >
259 THEN AR = 0
79 1560 IF AR < 0 THEN AR = 259
80 1570 AX = 2 * INT (AR / 2): I
F AE > AB THEN A = A + 1
: IF A > AE THEN A = AB
81 1580 IF AE < AB THEN A = A -
1: IF A < AE THEN A = AB
82 1590 GOTO 1530
83 1600 IF C < > 160 THEN 1630
84 1610 IF PEEK (49152) < 128 TH
EN 1610
85 1620 POKE 49168,0: RETURN
86 1630 IF C = 136 THEN GOSUB 16
70: GOTO 710
87 1640 IF C = 137 THEN GOSUB 16
60: GOTO 710
88 1650 GF = 1: RETURN
89 1660 AS = AS + (AS < 15): RET
URN
90 1670 AS = AS - (AS > - 15): R
ETURN
91 1680 AP = AP - (AP > 0): RETU
RN
92 1690 AP = AP + (AP < 150): RE
TURN
93 1700 DATA QUIT,DRAW,MOVE,ERAS
E,CLEAR,SAVE,INVERT,UPDA
TE,REVERT
94 1710 DATA BLACK1,GREEN,PURPLE
,WHITE1,BLACK2,ORANGE,BL
UE,WHITE2
95 1720 DATA EDIT FRAME,INSERT F
RAME,DELETE FRAME,RANGE
BOTTOM,RANGE TOP
96 1730 DATA ANIMATE,FASTER SPEED
0,SLOWER SPEED,LESS PAUSE
E
97 1740 DATA MORE PAUSE,LDA0,SAV
E,CLEAR,INVERT,QUIT

```

## Program 7: TI Animator

Version by Patrick Parrish,  
Programming Supervisor

```

10 FDR=2: BACK=12: REM
REQUIRES EXTENDED BASIC
20 CALL CLEAR: GDSUB 400
30 F1=1: L=10: N=1
40 BE=1: E=20: SP=0:
50 OL=0: FDR=10: TD 14
60 3: CALL CHAR(I,RPT$(0
,16)): NEXT I
70 CALL MAGNIFY(4): DIM B
(16,16),C$(15),E$(20),IFL
A$(20): G0="DEVICE (DSK

```

```

1.FILE OR C$1)?*:: GOSUB
B 910:: CALL SCREEN(BAC
K):: GOSUB 970
40 CALL KEY(0,K,S):: IF S=0
THEN 40
50 IF K>40 AND K<57 THEN K=
K-40:: ON K GOTO 170,50
0,220,220,240,240,250,25
0
60 IF K=66 THEN GOSUB 260
70 IF K=70 THEN GOSUB 270
80 IF K=73 THEN GOSUB 280
90 IF K=69 THEN GOSUB 300
100 IF K=65 THEN GOSUB 330
110 IF K=76 THEN GOSUB 370
120 IF K=67 THEN GOSUB 390
130 IF K=81 THEN END
140 IF K=79 THEN GOSUB 410
150 IF K=68 OR K=83 THEN N=
N-(N<0)-1-(N=1):19:(K=
83)+(N<20)-(N=20):19:
(K=68):: DISPLAY AT(4,
1):N:
160 GOTO 40
170 CALL CLEAR:: GOSUB 490
:: FOR I=5 TO 8:: CAL
L COLOR(I,FORE,1):: NEX
T I:: FOR I=1 TO 20::
CALL CHAR(60+I*4,E*(I)
):: NEXT I
180 GOSUB 490:: CALL SPRIT
E(0,60+BE*4,FORE,100,1
00):: CALL MOTION(0,0,
SP*6)
190 FOR I=BE TO E:: CALL P
ATTERN(0,60+I*4):: DIS
PLAY AT(1,I):: IF FOR J
=1 TO DL:: NEXT J:: N
EXT I
200 CALL KEY(0,K,S):: IF S=
0 THEN 190 ELSE CALL OE
LSPRITE(0):: CALL CHAR
SET
210 FORE=FORE-1:: GOSUB 27
0:: FOR I=5 TO 8:: CAL
L COLOR(1,2,1):: NEXT
I:: LF=1:: GOSUB 910
:: GOSUB 970:: GOTO 40
220 IF K=3 THEN BE=N ELSE E
=N
230 GOSUB 1060:: GOTO 40
240 SP=SP-(SP<0)-15:(K=5)+(
SP<0):15:(K=6):: GOSUB 1
060:: GOTO 40
250 DL=DL-(DL<0):(K=7)+(DL
=0):(K=7)+150+(DL<150)
:(K=8)-(DL=150):(K=8)+1
50:: GOSUB 1060:: GOT
O 40
260 BACK=BACK+1:(BACK=16):1
4:: CALL SCREEN(BACK)::
RETURN
270 FORE=FORE+1:(FORE=16):1
5:: FOR I=10 TO 14::
CALL COLOR(1,FORE,1)::
NEXT I:: CALL COLOR(9,
FORE,16):: RETURN
280 GOSUB 320:: IF A$="N"
THEN RETURN
290 FOR I=20 TO N+1 STEP -1
:: E*(I)=E*(I)-1:: NEX
T I:: E*(N)=RPT$(0,6
4):: GOSUB 450:: GOSUB
440:: RETURN
300 GOSUB 320:: IF A$="N"
THEN RETURN
310 FOR I=N TO 19:: E*(I)=
E*(I)+1:: NEXT I:: E*(
20)=RPT$(0,64):: GOSU
B 450:: GOSUB 440:: R
ETURN
320 DISPLAY AT(12,4)::ARE Y
OU SURE (Y/N)?:: ACCE
PT AT(12,24)VALIDATE(Y)

```



"Ti Animator" makes extensive use of redefined character graphics.

```

N)::A$:: IF A$="Y" THE
N RETURN ELSE GOSUB 440
:: RETURN
330 GOSUB 420:: OPEN #1:0=
INTERNAL,OUTPUT,FIXED
80:: PRINT #1:FORE::
PRINT #1:BACK:: FOR I=
1 TO 20:: PRINT #1:E*(
I)
340 PRINT #1:IFLAG(I):: NEX
T I
350 CLOSE #1:: FOR I=10 TO
14:: CALL COLOR(I,FOR
E,1):: NEXT I:: CALL S
CREEN(BACK)
360 IF ASC(0)=67 THEN GOSUB
B 970:: RETURN ELSE GO
SUB 440:: RETURN
370 GOSUB 420:: OPEN #1:0=
INTERNAL,INPUT,FIXED
80:: INPUT #1:FORE::
INPUT #1:BACK:: FOR I=
1 TO 20:: INPUT #1:E*(
I)
380 INPUT #1:IFLAG(I):: NEX
T I:: GOSUB 450:: GOT
O 350
390 GOSUB 320:: IF A$="N"
THEN RETURN
400 FOR I=1 TO 20:: E*(I)=
RPT$(0,64):: NEXT I::
GOSUB 450:: GOSUB 44
0:: RETURN
410 F=(F=1):10+(F=11):10
:: L=F+9:: GOSUB 450::
GOSUB 460:: RETURN
420 DISPLAY AT(12,1):0:: I
F 0$="N" THEN GOSUB 440
430 RETURN
440 CALL HCHAR(12,1,32,64)::
RETURN
450 J=0:: FOR I=F TO L::
CALL CHAR(104+J*4,E*(I)
):: J=J+1:: NEXT I::
RETURN
460 CALL HCHAR(9,2,32,2)::
J=F+1:: FOR I=2 TO 26
STEP 3:: DISPLAY AT(9,
I):J:: J=J+1:: NEXT I
:: F=STR$(F):: DISPLA
Y AT(4,1):N:
470 FOR I=1 TO LEN(F):: CA
LL HCHAR(9,1+ASC(SEG$(
F,I,1))):: NEXT I::
RETURN
480 DISPLAY AT(1,7)::PLEASE
WAIT...:: RETURN
490 CALL HCHAR(1,9,32,14)::
RETURN
500 DISPLAY AT(12,1)::TYPE
21 TO ABORT, A # 1-20,
OR <ENTER> FOR CURRENT
#
510 ACCEPT AT(13,27):A$::
IF A$="N" THEN PN=N:: G

```

```

OTO 500
520 IF A$<"1" OR A$>"9" THE
N 510
530 J=VAL(A$):: IF J=21 THE
N GOSUB 440:: GOTO 40
540 IF J>20 OR J<1 THEN 510
550 IF J=N THEN PN=N ELSE P
N=J
560 IFLAG(PN)=IFLAG(N):: CA
LL CLEAR:: DISPLAY AT(
2,1)::1-MOVE 2-DRAW
(4 SPACES)3-ERASE:: 0
ISPLAY AT(3,1)::4-CLEAR
5-INVERSE 6-0ISPLAY"
570 DISPLAY AT(4,1)::7-SAVE
8-QUIT:: 9-0ISPLAY AT
(5,2)::(USE ARROW KEYS
TO MOVE):: 0ISPLAY AT
(24,7)::FRAME::PN:: 0F
-1:: GOSUB 920
580 IF IFLAG(N)=1 THEN CALL
CHAR(100,US,101,"")ELS
E CALL CHAR(100,"",101,
U)
590 F$=E$(N):: FOR I=0 TO 4
STEP 2:: 0=ASC(SEG$(
F,I+1,1))-48:: 0=0+(0
>9):7:: 01=ASC(SEG$(F$
,I+2,1))-48:: 01=01+(0
1>9):7
600 0=-15*(IFLAG(PN)=1)+0*(
IFLAG(PN)=1)-(IFLAG(PN)
=0):0:: 01=-15*(IFLAG(
PN)=1)+01*(IFLAG(PN)=1)
-(IFLAG(PN)=0):0:01
610 DISPLAY AT(7+I/2+(I>3):
16,3-(I>3):8):C$(0)::
DISPLAY AT(7+I/2+(I>3)
):16,7-(I>3):8):C$(01)
:: NEXT I
620 CALL SPRITE(0,2B,96,14,4
9,33):: KHAR=101:: FLA
G=1:: R=1:: C=1
630 CALL KEY(0,K,S):: IF S=
0 THEN 630 ELSE CALL OE
LSPRITE(0)
640 IF K=51 THEN KHAR=100::
IF FLAG=0:: GOTO 790
650 IF K=49 THEN FLAG=1::
GOTO 800
660 IF K=50 THEN KHAR=101::
IF FLAG=0:: GOTO 790
670 IF K<52 THEN 690
680 FOR R=1 TO 16:: CALL H
CHAR(6+R,5,100,16):: NE
XT R:: OFLAG=0:: GOTO
620
690 IF K<53 THEN 720
700 OFLAG=0:: IFLAG(PN)=-(
IFLAG(PN)=0):: IF IFLAG
(PN)=0 THEN CALL CHAR(1
00,"",101,US)ELSE CALL
CHAR(100,US,101,"")
710 GOTO 800
720 IF K=54 THEN CALL OELSP
RITE(ALL):: LF=1:: GOS
UB 910:: GOSUB 970:: G
OTO 40
730 IF K=66 THEN GOSUB 260
:: OFLAG=1
740 IF K=78 THEN GOSUB 270
:: OFLAG=1
750 IF K=54 OR K=55 THEN CA
LL OELSPRITE(ALL):: IF
OFLAG=1 THEN 890 ELSE B
10
760 C=C+(K=83)+(C=1):(K=83
):16:(K=68)-(C=16):(K
=68):16)
770 R=R+(K=69)+(R=1):(K=69
):16:(K=88)-(R=16):(K
=88):16)
780 CALL LOCATE(0,2B,0,41,

```

```

00C+25): IF FLAG=1 THE
N 000
790 OFLAG=0 : CALL MCHAR(6
+R,4+C,KHAR)
800 CALL SOUND(20,200,5):
GOTO 630
810 GOSUB 400 : FOR R=1 TO
16 : FOR C=1 TO 16 :
CALL GCHAR(6+R,4+C,BC)
: GC=GC-100 : B(R,C)=
BC : NEXT C : NEXT R
820 F$="" : IF IFLAG(PN)=0
THEN HEX$="0123456789A
BCDEF" ELSE HEX$="FE0C
AB76543210"
830 FOR R=1 TO 16 : LOW=B(
R,5)+B(R,6)+B(R,7)+
2+B(R,8)+1
840 HIGH=B(R,1)+B(R,2)+B(
R,3)+B(R,4)+1
850 F$=F$SEG$(HEX$,HIGH,1)
&SEG$(HEX$,LOW,1): NEX
T R : FOR R=1 TO 16
860 LOW=B(R,13)+B(R,14)+
B(R,15)+B(R,16)+1
870 HIGH=B(R,9)+B(R,10)+
B(R,11)+B(R,12)+1
880 F$=F$SEG$(HEX$,HIGH,1)
&SEG$(HEX$,LOW,1): NEX
T R : IF K=54 THEN OFL
AG=1
890 IF K=55 THEN ES(PN)=F$
: LF=1 : GOSUB 910 :
GOSUB 970 : GOTO 40
900 CALL MAGNIFY(4): CALL
CHAR(36,F$): GOSUB 490
: CALL SPRITE(11,36,F
ORE,00,175): GOTO 620
910 CALL CHAR(33,RPT$("B1",
B),36,RPT$("0",14)+F$
,37,"FF&RPT$("0",14)+F$
,B,RPT$("01",B),39,RPT$
("00",B)): IF LF=1 THEN
LF=0 : RETURN
920 US=RPT$("F",16): CALL
CHAR(100,"",101,US,96,"
F&909F&RPT$("0",57)):
: CALL COLOR(9,FORE,16)
: IF SF=1 THEN SF=0 :
RETURN
930 F$="0000000100100011010
0010111001111000000110
10101110011011101111
940 FOR I=0 TO 15 : Z$=SEG
$(F$,I*4+1,4): D$=""
950 FOR J=1 TO 4 : T=VAL(S
EG$(Z$,J,1))+100 : O$=
O$&CHR$(T): NEXT J :
C$(I)=O$ : NEXT I
960 FOR I=1 TO 20 : E$(I)=
RPT$("0",64): NEXT I :
: RETURN
970 CALL CLEAR : DISPLAY A
T(1,10): "ANIMATOR" : C
ALL MCHAR(4,2,35): DIS
PLAY AT(4,5): "B"
(4 SPACES)E="(4 SPACES)B
P="(4 SPACES)D=""
980 GOSUB 1040 : CALL VCHA
R(4,1,30,2): CALL VCHA
R(6,31,39,2): FOR I=2
TO 29 STEP 3 : CALL HC
HAR(8,1,36,2) : CALL HC
HAR(8,1,37,2)
990 NEXT I : FOR I=4 TO 28
STEP 3 : CALL VCHAR(6
,1,33,2): NEXT I : GO
SUB 460 : GOSUB 450 :
J=104
1000 FOR I=2 TO 29 STEP 3 :
: CALL MCHAR(6,1,J):
CALL MCHAR(6,I+1,J+2):
: CALL MCHAR(7,1,J+1):
: CALL MCHAR(7,I+1,J+3)
: J=J+4 : NEXT I
1010 CALL MCHAR(15,1,49):
CALL MCHAR(16,1,52):
CALL MCHAR(17,1,55)
1020 DISPLAY AT(15,1): "ANIM
ATE 2 EDIT(3 SPACES)3
START PIC" : DISPLAY
AT(16,1): "END PIC 5 -S
PEED 4 +SPEED"
1030 DISPLAY AT(17,1): "SWI
TCH 8 +SWITCH"
1040 DISPLAY AT(20,1): "(I)N
SERT PIC D(0)LETE PIC
" : DISPLAY AT(21,1):
"S(A)VE (L)OAD (C)LE
AR ALL"
1050 DISPLAY AT(22,1): "(N)E
XT GROUP(4 SPACES)(Q)U
IT" : RETURN
1060 DISPLAY AT(4,7): BEJ:
DISPLAY AT(4,13): E:
DISPLAY AT(4,20): SP:
DISPLAY AT(4,26): STR$
(OL)&SEG$("&(3 SPACES)"
,1,3-LEN(STR$(OL)):
: RETURN

```

## Learn Computer Programming The ICS Way...



**T**rain at home in spare time! No previous experience needed! Now you can learn it all! Computer programming... computer applications... computer games... everything you ever wanted to know about computer operations! Write your own computer programs or use hundreds of programs already available... budgeting, real estate, bookkeeping, expenses, investments, interest, taxes, shopping lists, vacation planning, addresses, phone numbers, routing... even foreign languages and graphics.

You'll know what really happens when you have a computer problem with a book above your company or company utility or anyone else. You'll be able to talk their language, understand why and how things happen. You'll be able to take the offensive when you're the victim of a computer error. Experts explain everything in easy-to-understand language with step-by-step directions. Computer included with your training... plug into your TV. **SEND FOR FREE FACTS!**

**ICS COMPUTER TRAINING, Dept. 82275**  
Scranton, Pennsylvania 18515

Push the free facts that tell how I can learn computer applications, programming and operation at home in spare time. No cost. No obligation. No salesman will visit you!

Name  Age   
Address   
City/State/Zip   
Phone No. (  )

## We wrote the book on the ATARI-ST

First there was the fabulously successful VIC-20. Then came the record-breaking Commodore-64.

Now Jack Tramiel has launched his third home computer, the ATARI ST. The ST promises to shatter all exciting price-performance barriers and become a leader in the home-computer market.

This book, **PRESENTING THE ATARI ST** gives you an in-depth look at this sensational new computer that promises to bring you... "Power without the Price." Some of the topics include: • History of Atari • Overview of the ST • Construction, Operating System • Peripherals • Languages • User Programs, and more.

**PRICE \$16.95** TO ORDER: SEND CHECK, MONEY ORDER OR CHARGE NO. TO:

**Abacus Software**

P.O. BOX 7211 GRAND RAPIDS, MI 49510 - Fast Service, Phone 616-241-5510



A BACA BENTON BOOK PUBLISHED BY



# Archive: Two-Drive Backup for Commodore 64

Philip I. Nelson, Assistant Editor

*Now you can copy entire disks at machine language speed with this convenient backup program for the Commodore 64 with two 1541 disk drives. It also works on the new Commodore 128 in 64 mode.*

Sooner or later it's bound to happen. You'll make an unconscious error, or lightning may strike while you're resaving a program, or the family dog will chew a few disks for dinner—and an important disk will be utterly destroyed. If you have a backup copy, of course, such accidents aren't fatal. You take a moment to pat yourself on the back, pull out the archive disk, and go back to work.

If you don't have a backup, it's like watching a gold ring slip off your finger and go clanking down the drain. In the long hours spent reconstructing what you've lost, you have plenty of time to reflect on the wisdom of archiving your work on a regular basis.

Archiving is one of those grim tasks that's easy to postpone. BASIC programs (like COPY/ALL on your 1541 Test/Demo disk) are slow, and may not copy machine language (ML) programs or sequential files. Even good single-drive backup programs keep you tied to the computer, tapping your fingers until it's time for the next disk swap.

## Fast And Convenient

"Archive" offers a better way: It links two drives together to take the

misery out of backing up important disks. To speed things up, it's written entirely in machine language and copies only those disk sectors which actually contain data. But because it loads and runs just like a BASIC program, it's easy for anyone to use, even beginners.

You may find this program valuable even if you don't own two disk drives. Put your drive together with a friend's and swap several disks during one session. Or bring it to a user group meeting to speed up the duplication of public domain library disks. Since 1541-format disks work with other Commodore computers, Archive running on a Commodore 64 can also copy disks that will be used with the Commodore 128, VIC-20, Plus/4, 16, and 4040-format PET/CBM. (Of course, a program written for one of these machines may not work on another. Also, Archive cannot copy Commodore 128 CP/M disks.)

Archive has been tested successfully on the Commodore 128 in 64 mode with two 1541 drives. If the new 1571 drives are truly 1541-compatible, Archive will work with them as well, since it uses standard Commodore disk commands. However, the 1571 was not available for testing when this article was written.

Incidentally, Archive cannot duplicate commercially protected software. Protected disks invariably contain deliberate errors (which shut down the program) or data hidden in unused sectors (which Archive does not copy).

## Getting Started

Enter and save Archive using the MLX machine language entry program published elsewhere in this issue. Here's the information you need:

Starting address: 49152

Ending address: 51185

After you save Archive, activate it like a BASIC program by typing LOAD"ARCHIVE",8 followed by RUN. (For this program, do not use ,8,1 after the LOAD; just use ,8.) If you're already comfortable using two drives, you needn't read any further, since Archive prompts you at each step. Just pop a disk in each drive as instructed, press the f7 special function key, and relax while Archive does its work. (If you've never used two drives before, see "Setting Up Your System" below.)

Archive displays your source disk's Block Availability Map (BAM) graphically on the screen, updating the display as copying proceeds. Thus, you can tell at a glance how much of the disk is used and how much has been copied. The number at the lower right of the screen shows the sector being copied; the graphic display shows which sectors have already been copied.

If you want to abort the copy for any reason, press the f1 special function key to return to BASIC. (When you abort the copy process, the archive disk is incomplete and may be garbled. You can reuse it immediately with Archive, but do not use it for anything else without reformatting it as explained below.)



Once the copy is done, press the f3 function key to copy another disk, or press f1 to quit. Whenever you exit Archive, it clears the screen and reports the status of each drive.

### Quick Formatting

Since Archive always makes a complete disk copy, it *formats* the archive disk with a NEW command. Formatting renames the disk and erases everything it contained

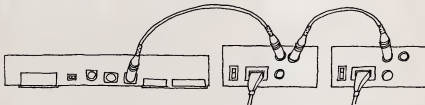
Figure 1 shows how to daisy-chain two drives to your computer. Connect the first drive as usual, then plug the serial cable from the second drive into the extra serial port connector on the first drive.

When more than one disk drive is active, each drive must be given a different *device number* so the computer can tell them apart. The 1541 is factory-set as device 8, but it can also have device numbers

```
OPEN 15,8,15
PRINT#15,"M-W"CHR$(119)
CHR$(0)CHR$(2)CHR$(32+9)
CHR$(64+9)
CLOSE15
```

2. It's a good idea to verify the device number change. Put a disk in the drive, then type LOAD"\$",9 and press RETURN to load its directory. After the blinking cursor returns, type LIST and press RETURN. If you see the directory, the

### Daisy-Chaining Two Disk Drives



The second drive is plugged into the first drive's extra serial port connector (rear view).

before. You'll notice that the archive disk is formatted in only a few seconds rather than the usual couple of minutes, and without the usual knocking sound. To save time and minimize wear on the drive, Archive uses a shortened NEW command: the equivalent of OPEN 15,8,15,"N0:filename" without a disk ID.

The abbreviated NEW command works only on a disk that has been previously formatted. To use a brand new disk, you must prepare it first with a full NEW command: OPEN 15,8,15,"N0:filename,ID". The filename can be up to 16 characters long. The ID can be any two letters or numbers and should be unique for each disk. The 1541 User's Manual contains more information about formatting disks.

### Setting Up Your System

Although the Commodore 64 has only one serial port connector, the 1541 disk drive has two, letting you hook up more than one drive at a time. Since the drives are chained together in a series, this arrangement is often called *daisy-chaining*.

9-15. Archive uses device numbers 8 and 9, reading from drive 8 and copying to drive 9. You must always put the source disk (the original) in the drive that's device 8 and the archive disk (the copy) in the drive that's device 9.

If both of your drives are device 8, don't despair. You can easily change one of them to device 9. The change is temporary; the drive reverts to device 8 when you turn off the power. Here's the procedure:

1. Turn on the drive that you want to change to device 9. Make sure the other drive is turned off. Now you can change the device number either by running the DISK ADDR CHANGE utility program on your 1541 Test/Demo disk, or by typing in direct statements.

To use DISK ADDR CHANGE, load the program from the 1541 Test/Demo disk and enter RUN. Follow the program's instructions, then skip to Step 2 below.

You can also change the device number by entering the following statements in direct mode (with no line numbers). Press RETURN after you type each line:

change worked and you may proceed to step 3. If you get an error (probably ?DEVICE NOT PRESENT), turn off the drive and repeat step 1.

3. Turn on the other drive. This drive will remain device 8 (the source drive). Now load and run Archive, inserting the disks as explained in the instructions. The source (original) disk goes in device 8, and the archive (copy) disk goes in device 9. As an additional precaution, you may want to write-protect the source disk by taping over the notch in the sleeve.

In theory you can daisy-chain several drives to a 64, but in fact the 1541 doesn't enjoy sharing the serial bus. The drives should always be turned on one at a time, not simultaneously (as would happen with a power strip). Printer interfaces that draw power from the 64's cassette port are notorious for causing disk errors, and other peripherals can affect system voltage levels even if they're not turned on. Depending on your system, you may need to unplug other peripherals before using Archive.

# Trusted Software and

## LANGUAGE SOFTWARE

### ADA TRAINING COURSE

Teaches you the language of the future. Comprehensive subset of language. Includes: editor; syntax checker; compiler; assembler; disassembler; handbook. \$59.95



### ASSEMBLER/MONITOR-64

Macro assembler and extended monitor. Supports all standard functions plus floating point constants. Monitor supports bank switching, quick trace, single step, more. \$39.95



### BASIC COMPILER 64

Compiles complete BASIC language into either fast 6510 machine language and/or compact speedcode. Supports overlays and many extended BASIC commands. \$39.95



### C LANGUAGE COMPILER

Full compiler per Kernighan & Ritchie standard, but without bit fields. Includes powerful editor (41K source file); compiler (6510 code), library (supports many functions) and linker. NEW! \$79.95



### FORTH-64 LANGUAGE

Based on Forth 79 (+ parts of '83). Supports hires graphics and sound synthesizer. Full screen editor, programming tools, assembler, simples, handbook. NEW! \$39.95



### MASTER-64

Professional development package for serious applications. Indexed file system, full screen & printer management, programmer's aid, multiprecision math, royalty-free runtime, more. \$39.95



### PASCAL-64

Full Pascal supports graphics, sprites, file management, more. Supports pointers, dynamic memory management, machine language. Compiles to fast 6510 machine code. \$39.95



### VIDEO BASIC-64

Add 50+ graphic, sound and utility commands to your programs with this super development package. Free distribution of RUNTIME version - no royalties! \$59.95



**XREF-64** Basic cross-reference indispensable tool for BASIC programmer. Cross-references all references to variables, constants and line numbers. Sorts them into alphabetical order for convenient use. \$17.95



## FOR COMMODORE COMPUTERS

## REFERENCE BOOKS



**ANATOMY OF C-64** Insider's guide to the '64 insens. Graphics, sound, I/O, kernel, memory maps, more. Complete commented ROM listings. 300pp \$19.95

**ANATOMY OF 1541 DRIVE** Best handbook on floppy explains all. Many examples and utilities. Fully commented 1541 ROM listings. 320pp \$19.95

**MACHINE LANGUAGE C-64** Learn 6510 code, write fast programs. Many examples & listings for complete assembler, monitor, simulator. 200pp \$14.95

**GRAPHICS BOOK C-64** - best reference covers basic and advanced graphics. Sprites, animation, Hires, Multicolor, lightpen, 3D-graphics, IRQ, CAD, projections, curves, more. 350pp \$19.95

**TRICKS & TIPS FOR C-64** Collection of easy-to-use techniques: advanced graphics, improved data base, enhanced BASIC, CPM, more. 275pp \$19.95

**1541 REPAIR & MAINTENANCE** Handbook describes the disk drive hardware. Includes schematics and techniques to keep 1541 running. 200pp \$19.95

**ADVANCED MACHINE LANGUAGE** Not covered elsewhere - video controller, interrupts, timers, clocks, I/O, real time, extended BASIC, more. 210pp \$14.95

**PRINTER BOOK C-64/VIC-20** Understand Commodore; Epson-compatible printers, 1520-plotter. Packed: utilities; graphics dump, 3D-plot; commented MPS601 ROM listings, more. 330pp \$19.95

## PRODUCTIVITY TOOLS

### CHARTPAK-64

#### MAKE YOUR OWN CHARTS...

Professional quality charts and graphs instantly. Easy-to-use menus & chart formats. Two size hardcopy. Accepts data from Multiple CalcResult. Supports statistical functions. \$39.95  
Also available  
**CHARTPLOT-1520** fine output to inexpensive 1520 plotter. \$39.95  
**CHARTPLOT-64** unsurpassed quality charts on most XY-plotters. \$84.95



### POWER PLAN-64

POWER PLAN is our top-rated, full-featured spreadsheet for both your business and personal needs. All menu selections are clearly displayed on screen. Built-in HELP screens guide you through this powerful package. Immediately make 2D or 3D pie, bar, line and point to point graphs from your spreadsheet using up to 8 windows. Features cell protection and formatting, split screens, all major math functions, 25K worksheet space, 255 rows X 63 columns. 200 page handbook. \$49.95

#### Super Spreadsheet with Graphs



### CADPAK-64

#### DETAIL YOUR DESIGNS...

Superb lightpen design tool. Exact placement using *Auto-Point* positioning. 2 complete screens; draw LINES, BOXES, CIRCLES, ELLIPSES; pattern FILLing; freehand DRAW, screen COPY; ZOOM for detail. Two size hard-copy. Requires good quality lightpen. \$49.95



**FROM ABACUS SOFTWARE  
... A NAME YOU CAN COUNT ON**



For your convenience, the programs contained in each of our books are available on diskette to save you time entering them from your keyboard. Specify name of book when ordering. **\$14.95 each**

**SCIENCE/ENGINEERING ON C-64**  
In depth intro to computers in science.  
Topics: chemistry, physics, biology, astro-  
nomy, electronics, others. 350pg \$19.95

**CASSETTE BOOK C-64/VIC-20**  
Comprehensive guide; many sample programs. High speed operating system-fast file loading and saving. 225pp. \$14.95

**IDEAS FOR USE ON C-64 Themes:** auto expenses, calculator, recipe file, stock lists, diet planner, window advertising, others. Includes listings. 200pp \$12.95

**COMPILER BOOK C-64/C-128** All you need to know about compilers: how they work; designing and writing your own; generating machine code. With working example compilers. **300pp \$19.95**

**Other Books Coming Soon!!**

**PEEK & POKES** - Programming quickies that will amaze you. Includes in-depth explanations of PEEK, POKE, USR and other commands. **200pp \$14.95**

**ADVENTURE GAMEWRITER'S Handbook**  
Step-by-step guide to designing and writing your own adventure games. With automated adventure game generator. 200pp \$14.95

**CAD BOOK for C-64/C-128** Introduction to computer aided design. 2D and 3D, reflection, zooming, macros. Samples use SIMON's Basic. 250pp \$19.95

**MORE TRICKS & TIPS** Dozens of more helpful programming techniques: software protection, new BASIC commands, interrupts; ROM routines; the KERNAL, hardware tips, etc. 250pp \$19.95

## CHART YOUR OWN STOCKS.

Sophisticated technical analysis charting package for serious investor. Capture data from DJN/RS or Warner Services or enter and edit your data. 7 moving averages, 3 oscillators, trading bands, least squares, 5 volume indicators, relative charts, much more. Two size hardcopy. **\$84.95**



## ORGANIZE YOUR DATA

Powerful, easy-to-use data management package uses menu selections. Free form design: 50 fields/record; 2000 records/disk. Sort on multiple fields in any combination. Complete selection and formatting for existing reports. **\$39.95**

TEXTOMAT-64

WORD PROCESSING...

Flexible wordprocessor displays 40 or 80 columns with horizontal scrolling. Menu selections make TEXTOMAT easy to use. Quickly move from editing to formatting to merging to utilities. Supports virtually any printer. Can support foreign languages. **\$39.95**

QUICKCOPY V2.0

PROTECT Your DATA ...

Backup your valuable data with the fastest disk copier we've seen to date. Copies entire disk in 2.5 minutes on two or 3.5 minutes on one. 1541. Very useful utility. **\$19.95**



XPER - Software that Learns



XPER is the first expert system for the C-64 and C-128. Ordinary data bases are good for reproducing facts, but XPER lets you derive knowledge from a mountain of facts and helps you make expert decisions. Scientists, doctors and professionals are currently using XPER in their research projects.

The XPER system puts your information into a *knowledge base*. A knowledge base consists of a variable number of objects - such as trees; **features** - such as "what kind of leaf does it have?"; and **attributes** - such as 6 inch needles. After describing your knowledge base, you can easily enter your information using XPER's complete entry/maintenance facilities.

Then, XPER's very efficient searching techniques will guide you through the most complex decision making criteria. You'll also benefit from XPER's full reporting capabilities and ease of use.

## ORDERING INFORMATION

Call now for the name of your nearest dealer

# Abacus Software

P.O. Box 7211 Grand Rapids, Michigan 49510



For postage and handling include \$4.00 per order. Foreign orders include \$8.00 per item. Money order and checks in U.S. Dollars only. Mastercard, VISA and American Express accepted. Michigan residents please include 4% sales tax.

For fast service call (616) 241-5510 Telex 709-101

For **free** catalog, please return this coupon to:

Abacus Software, P.O. Box 7211, Grand Rapids, MI 49510

**PHONE: (610) 344-5510**

**PHONE: (616) 241-5510**

Name \_\_\_\_\_

Address

City

State Zip

• Compute SWS

## Initialization

Before it starts copying, Archive initializes each disk to test whether devices 8 and 9 are active and if each contains a formatted disk. The initialization command transfers information (disk name, ID, etc.) from the disk into the drive's memory to prepare it for handling the disk.

If this step fails, it means one of the drives is not ready to go. Archive displays the status of both drives and returns you to BASIC. (If you forget to change one of the drives to device 9, Archive can't read its status; press RUN/STOP-RESTORE and proceed as explained below.) Enter these lines in direct mode (without a line number) to retry the initialization:

```
OPEN 15,15,"0":CLOSE 15
OPEN 15,15,"0":CLOSE 15
```

When you enter each line, the drive motor should run and its red light should glow. After one or two seconds the red light should go off and stay off, and you should be able to run Archive. If one or both of the red lights blink continuously, turn off both drives and repeat the setup process. The drive makes a knocking sound if you forget to insert a disk or try to use an unformatted disk for the archive.

## Is Your Drive Healthy?

In ordinary use your drive works intermittently. It may spend 30 seconds loading a game for you, then sit idle for 30 minutes while you play. Copying a full disk with Archive is far more demanding work, requiring several minutes of continuous running. If one of your drives is misaligned, tends to overheat, or has other mechanical problems, don't be surprised if you experience occasional errors. When an error occurs during the copy process, Archive stops copying, reports the status of both drives, and returns to BASIC.

Such errors are especially likely to crop up when the source disk is nearly full. For mechanical reasons it's harder for the drive to access the disk's outer area than the area near the middle. To make things easy on itself, the drive always starts storing programs in the middle of the disk, leaving the outer tracks empty until there's no

room left elsewhere.

Archive's BAM display lets you observe this storage scheme. When the source disk contains only a few programs, they'll all be stored in middle tracks (near track 18). The outermost tracks (1 and 35) are usually the last to be used. If your drive consistently has trouble accessing outer tracks, it's probably misaligned. The same problem can result if the disk was formatted on a badly misaligned drive.

## 1541 ML Programming

To shorten and speed up the program, all of Archive's 21 variables and pointers are located in the zero page (lowest 256 bytes) of memory. Zero page machine language instructions run faster and use less memory than instructions that reference higher memory addresses. The computer can find what it needs by checking only one byte, rather than wading through a two-byte address in search of the same information. In time-critical programs like Archive, which execute certain routines many thousands of times a minute, the microseconds you save can add up to a significant difference in running time.

Many programmers have trouble learning to handle disk files in machine language. For those who are interested, here's an outline of Archive's main routines.

```
50052-0063 Initialize device 8
50064-0078 Initialize device 9
50079-0018 Error-report status
50019-0061 Read BAM from source disk
50062-0A32 Display BAM and disk name
50A33-0A7E Short New destination disk
50A7F-0A8F OPEN 3,15 command channel
50A90-0A0A OPEN 5,15 command channel
50AA1-0A01 OPEN 4,4,"F" buffer channel
50A85-0A03 OPEN 6,6,"F" buffer channel
50A86-0A03 Subroutine-copy 3 block
50A87-0C04 Read block from source disk
50C05-0C42 Write block to archive disk
50C43-0C3C Subroutine-initialize disk
50C04-0C4A Subroutine-check error channel
50C04-0C3B String-BAM Block Read (U)
50C0C-0C3F String-buffer-shoot NEW
50C3E String-"F" for buffer channel
50C3E-0C2E String-"0" to initialize
50004-070F String-buffer-Block Read (U)
5070F-0718 String-Block Write (U)
50FF3 256-byte data buffer starts here
```

First the program maps the source disk's BAM on the screen to record which sectors contain data. Then it copies each used sector in turn, reading it from the source disk and writing it to the archive disk. Note that to read a disk sector, you

should always use the U1 direct access command rather than B-R (Block Read). Likewise, the U2 command must be used in place of B-W (Block Write). Despite what your user's guide says, the B-R and B-W commands are defective and should never be used.

## Archive: Two-Drive Backup

Please refer to the "MLK" article in this issue before entering the following listing.

```
49152 :027,000,000,000,158,050,243
49158 :040,055,055,055,143,034,143
49164 :020,020,020,020,020,032,144
49170 :065,062,067,072,073,086,207
49176 :069,080,080,080,162,015,014
49182 :142,033,200,142,032,200,027
49188 :337,143,130,000,169,028,231
49194 :160,015,032,030,171,169,107
49200 :124,133,176,133,170,169,193
49206 :084,133,177,169,216,133,118
49212 :179,165,197,201,084,240,022
49218 :086,201,083,208,246,240,202
49224 :086,169,089,032,118,255,243
49230 :076,121,080,169,227,160,071
49236 :112,032,030,171,169,000,250
49242 :131,002,032,067,012,165,245
49248 :144,040,021,169,000,160,134
49254 :013,032,030,171,169,000,014
49260 :133,002,032,067,012,165,007
49266 :144,040,003,076,025,009,163
49272 :169,003,032,195,255,169,175
49278 :084,032,195,255,169,005,018
49284 :032,195,255,169,000,032,053
49290 :195,255,169,013,032,195,229
49296 :169,013,032,195,255,041
49302 :032,204,255,169,206,160,152
49308 :015,032,030,171,169,015,076
49314 :160,162,000,032,160,255,205
49320 :169,000,032,109,255,077,077
49326 :192,255,162,015,032,190,004
49332 :255,032,207,255,201,013,119
49338 :240,014,201,032,240,004,149
49344 :201,065,144,241,032,210,005
49350 :255,076,102,000,169,013,135
49356 :032,195,255,032,204,255,153
49362 :032,231,255,169,224,160,001
49368 :015,032,030,171,169,015,136
49374 :160,162,009,032,186,255,018
49380 :169,000,032,109,255,032,137
49386 :192,255,162,015,032,190,004
49392 :155,032,207,255,201,013,179
49398 :140,014,201,032,240,004,209
49404 :201,065,144,241,032,210,21
49410 :255,076,242,000,169,015,255
49416 :032,195,255,032,204,255,213
49422 :169,013,032,210,255,162,007
49428 :128,100,000,000,169,007,003
49434 :160,013,032,030,171,169,009
49440 :015,160,162,000,032,186,091
49446 :255,162,013,032,189,182
49452 :032,192,255,169,013,160,105
49458 :162,000,032,186,255,169,034
49464 :001,162,224,160,012,032,135
49470 :189,255,032,192,255,162,123
49476 :015,032,201,255,162,000,221
49482 :189,192,012,032,210,255,196
49488 :222,224,013,206,245,032,010
49494 :255,162,013,032,189,182
49500 :255,162,000,032,207,255,235
49506 :157,243,015,232,200,247,176
49512 :032,204,255,162,015,032,036
49518 :148,012,165,167,240,003,077
49524 :186,121,000,169,013,032,023
49530 :175,255,169,015,032,195,215
49536 :222,224,013,160,013,032,112
49542 :030,171,169,013,032,195,252
49548 :255,162,144,189,243,013,124
49554 :032,210,255,232,224,164,239
49560 :200,245,169,013,032,210,005
49566 :255,169,146,160,013,032,165
49572 :100,171,160,000,182,000,161
49578 :169,004,132,165,165,176,214
49584 :133,251,165,177,133,252,007
```

49598 165,178,133,083,165,179,237  
49599 133,084,832,165,162,162,184  
49600 169,169,181,133,085,165,163  
49601 166,837,085,248,086,169,055  
49602 184,145,251,208,084,169,085  
49603 181,145,251,832,893,812,858  
49604 282,208,234,832,165,812,847  
49605 162,088,169,081,133,085,190  
49606 165,166,837,085,248,086,181  
49607 165,166,145,251,208,084,835  
49608 169,081,145,251,832,893,245  
49609 162,208,234,832,165,877  
49610 162,162,085,169,081,133,224  
49611 165,166,837,085,248,118  
49612 169,166,145,251,208,086  
49613 164,169,081,145,251,832,186  
49614 893,812,202,208,234,238,233  
49615 165,238,176,238,178,238,213  
49616 165,165,806,201,835,176,111  
49617 802,144,131,169,085,168,198  
49618 164,832,836,171,168,144,085  
49619 162,088,185,243,815,157,046  
49620 812,832,238,238,238,816,181  
49621 208,244,832,284,255,169,152  
49622 151,168,162,089,832,186,130  
49623 255,169,080,832,189,255,208  
49624 832,192,255,162,815,832,082  
49625 201,255,162,088,189,204,875  
49626 812,832,218,255,232,224,835  
49627 812,208,245,832,204,255,839  
49628 165,166,837,085,248,118  
49629 167,248,083,876,12,080,215  
49630 169,815,832,195,255,832,848  
49631 204,255,169,083,162,088,157  
49632 168,815,832,186,255,169,179  
49633 802,832,189,255,832,192,868  
49634 255,169,085,162,089,168,134  
49635 815,832,186,255,169,088,837  
49636 189,255,832,192,255,805  
49637 169,815,832,186,162,088,139  
49638 186,255,169,081,162,22,139  
49639 168,812,832,189,255,832,084  
49640 192,255,169,086,168,162,106  
49641 805,832,186,255,169,081,868  
49642 162,224,168,812,832,189,281  
49643 255,832,192,255,168,086,866  
49644 832,086,169,084,133,165,843  
49645 169,081,133,249,169,124,829  
49646 133,176,169,084,133,177,238  
49647 169,081,133,258,165,176,809  
49648 133,251,165,177,133,257,809  
49649 162,815,177,251,201,896  
49650 208,817,165,197,201,884,806  
49651 208,802,248,891,832,148,197  
49652 811,165,167,248,802,208,819  
49653 802,832,126,812,202,208,158  
49654 227,165,249,201,818,144,242  
49655 814,201,825,144,814,201,899  
49656 831,144,814,201,836,144,876  
49657 144,248,856,162,089,208,197  
49658 818,168,081,208,806,162,809  
49659 802,802,802,802,802,802,802  
49660 251,201,801,208,817,165,197  
49661 157,201,804,208,802,248,132  
49662 248,832,148,811,165,167,893  
49663 248,802,208,819,832,126,175  
49664 812,282,208,227,238,176,897  
49665 228,249,165,249,281,836,178  
49666 248,803,876,221,838,169,829  
49667 804,832,195,255,168,080,233  
49668 832,195,255,169,083,832,888  
49669 195,255,169,085,832,195,179  
49670 255,832,204,255,165,167,156  
49671 208,831,169,088,133,082,147  
49672 832,867,812,169,089,133,824  
49673 802,832,867,812,169,844,198  
49674 168,815,832,836,171,165,185  
49675 197,201,801,248,802,201,215  
49676 804,280,846,876,121,080,813  
49677 876,828,088,152,872,138,187  
49678 872,165,249,832,175,812,887  
49679 165,169,141,811,815,141,838  
49680 832,815,141,149,807,165,158  
49681 178,141,812,815,141,824,159  
49682 815,141,158,807,165,258,144  
49683 832,178,812,165,141,186  
49684 814,815,141,826,815,141,826  
49685 819,887,165,178,141,815,111  
49686 185,141,827,815,141,190,215  
50124 807,832,254,237,162,083,131  
50125 832,201,255,162,080,189,255  
50126 804,815,832,218,255,232,196  
50127 124,812,208,245,832,254,173  
50128 237,162,804,832,198,255,892  
50129 262,808,832,179,244,157,231  
50130 243,815,232,208,247,832,193  
50131 254,237,162,083,832,148,058  
50132 812,165,167,248,083,876,147  
50133 809,812,832,254,237,162,246  
50134 806,832,201,255,162,081,153  
50135 189,243,815,832,211,237,183  
50136 278,208,247,189,243,815,138  
50137 821,237,832,254,237,815,138  
50138 162,085,832,201,255,162,081  
50139 808,189,816,815,832,218,244  
50140 255,232,224,812,208,245,196  
50141 832,284,255,162,085,832,228  
50142 148,812,832,204,255,184,843  
50143 178,184,168,896,189,815,816  
50144 168,168,802,832,186,255,189  
50145 169,802,162,225,168,812,866  
50146 832,189,255,832,192,255,811  
50147 169,815,832,195,255,896,888  
50148 169,080,145,803,824,165,806  
50149 251,185,848,133,251,165,819  
50150 252,185,808,133,252,824,182  
50151 805,883,185,848,133,803,847  
50152 165,804,185,808,133,804,815  
50153 806,805,896,177,251,089,154  
50154 128,145,251,238,258,824,132  
50155 165,251,189,848,133,251,855  
50156 162,182,189,808,133,252,822  
50157 169,808,133,167,832,231  
50158 190,255,832,287,255,801,828  
50159 808,144,802,238,167,896,879  
50160 138,165,166,165,189,243,842  
50161 815,133,166,896,162,848,806  
50162 856,233,818,144,803,232,806  
50163 176,249,858,058,134,169,849  
50164 178,896,085,848,832,848,899  
50165 849,851,832,848,832,848,899  
50166 856,832,848,878,848,858,088  
50167 832,832,832,832,832,832,142  
50168 832,832,832,832,832,832,148  
50169 832,832,832,832,832,832,157  
50170 871,848,813,812,832,832,179  
50171 871,871,871,871,871,871,871  
50172 871,871,871,871,871,871,185  
50173 871,871,871,871,871,871,185  
50174 871,871,871,871,871,871,185  
50175 871,871,871,871,871,871,185  
50176 871,871,871,871,871,871,185  
50177 871,871,871,871,871,871,185  
50178 871,871,871,871,871,871,185  
50179 871,871,871,871,871,871,185  
50180 871,871,871,871,871,871,185  
50181 871,871,871,871,871,871,185  
50182 871,871,871,871,871,871,185  
50183 871,871,871,871,871,871,185  
50184 871,871,871,871,871,871,185  
50185 871,871,871,871,871,871,185  
50186 871,871,871,871,871,871,185  
50187 871,871,871,871,871,871,185  
50188 871,871,871,871,871,871,185  
50189 871,871,871,871,871,871,185  
50190 871,871,871,871,871,871,185  
50191 871,871,871,871,871,871,185  
50192 871,871,871,871,871,871,185  
50193 871,871,871,871,871,871,185  
50194 871,871,871,871,871,871,185  
50195 871,871,871,871,871,871,185  
50196 871,871,871,871,871,871,185  
50197 871,871,871,871,871,871,185  
50198 871,871,871,871,871,871,185  
50199 871,871,871,871,871,871,185  
50200 871,871,871,871,871,871,185

# Atari Color Mixing

Karl E. Wiegner

*This informative tutorial demonstrates the principles of color mixing on Atari 400/800, XL, and XE computers.*

Advertisements for home computers often tout the number of different colors that a particular machine can produce. But little is said about how these colors are generated on your TV or monitor screen. Knowing some theory behind these techniques can help you produce more colorful graphics displays.

Atari computers manufactured after early 1982 can generate 256 color variations, based on 16 different hues which each can have 16 luminances (brightnesses). Atari's made before early 1982 can display only 8 luminances per hue for a total of 128 colors, but can be upgraded by replacing the CTIA chip with a GTIA chip. Atari colors are represented by a number from 0-255 using this formula:  $\text{color number} = \text{hue number} * 16 + \text{luminance number}$ .

The "Atari Color Mixing" program listed below demonstrates additive color mixing and lets you try your hand at subtractive color mixing as well. We'll explain these terms in detail in a moment. For now, enter and save Color Mixing, then run it. The program is quite simple and contains all the instructions you need. Just follow the prompts and refer to the rest of this article for additional information.

The colors may look better if you adjust your TV's brightness control somewhat higher than usual. The exact hues may also vary depending on the tint setting. Adjust the tint for good green and red,

and the other colors should be pretty close.

## Additive Color Mixing

As every child discovers when painting with watercolors, the three primary colors—red, green, and blue—can be added together in various combinations to make many different colors. Thus, red, green, and blue are known as the *additive primary colors*.

When primary colors are combined, new colors appear. White light is a balanced combination of red, green, and blue light. Equal intensities of blue and green light produce a greenish-blue color called cyan. Red and blue light mix to create magenta, a reddish-purple. And, believe it or not, mixing red and green light together produces yellow. Many more colors can be produced if the primaries are of different intensities. For instance, combining a given intensity of green with twice that intensity of red is equivalent to yellow plus red, or orange.

Additive color mixing works with pigments as well as lights. However, pigment mixing often results in different colors than those described above. For example, mixing red and green paints usually produces brown, not yellow.

In fact, the brown color really is a yellow. But red and green pigments usually have far less intensity than red and green lights. Besides having a particular hue (light frequency), a color can have different levels of luminance (intensity). The effect we usually call color is actually the combined effects of hue and luminance. Because red and green pigments are quite dark

(have little luminance), mixing them together produces the dark yellow we commonly call brown.

A color TV or monitor creates different colors by the additive process using colored light sources. The screens of color TV tubes contain thousands of tiny red, green, and blue dots (phosphors) which glow when struck by electrons from a gun at the back of the tube. If you examine a color screen with a magnifying glass, you'll see individual red, green, and blue phosphors. At normal viewing distances the colored dots merge together and create additive colors. For instance, adjacent red and blue dots look like magenta. When no phosphors are lit, the result is black.

## Subtractive Color Mixing

Recall that if red and blue lights are combined, the resulting color is magenta. There's also another way to produce magenta—you can shine white light through a magenta-colored filter. White light contains all the primary colors, but the filter absorbs the green light, allowing only the red and blue light to pass. In other words, a magenta filter *subtracts* or blocks out green light. If you place a green filter and a magenta filter in front of a white light source, all light should be blocked out: The green filter blocks red and blue, and the magenta filter blocks green. The final result is black. For this reason, green and magenta are termed *complementary colors* (magenta is also sometimes called *minus green*).

Similar logic applies to the other primary colors: A cyan (blue + green) filter subtracts red light, and a yellow (red + green) filter sub-

tracts blue light. Red and cyan (minus red) are complementary, as are blue and yellow (minus blue).

Cyan, magenta, and yellow are called *subtractive primary colors*. Just like the additive primaries (red, green, blue), the subtractive primaries can be mixed into virtually any combination of hue and luminance. But the process is reversed. Additive color mixing works by sending specific colors to your eye, while the subtractive process removes specific colors from a color-rich light source, leaving only complementary colors. Most color photographic systems are subtractive, using cyan, magenta, and yellow film dyes.

Color Mixing uses color numbers which—on my system—come closest to producing the six additive and subtractive primary colors (see lines 270–280). Of course, colors can vary greatly from one TV or monitor to the next. Cyan is a little difficult to display; my choice for cyan would look a bit greener, but my computer won't cooperate. The blue is also darker than you might expect, but blue in a color mixing sense is actually quite dark.

## Additional Techniques

Note that each display screen in Color Mixing uses several different Atari graphics modes. The heading is displayed in graphics mode 1, other text is in mode 0, and the color squares are drawn in mode 3. Mixed-mode screens like this are created by modifying the computer's display list, a set of instructions which tells the computer how to put data on the screen. The Color Mixing program modifies display lists in lines 210, 1010–1020, 1210–1240, and 4010–4020. (You can read more about modifying display lists in *COMPUTE!'s First Book of Atari* and *COMPUTE!'s First Book of Atari Graphics*.)

Atari computers can ordinarily display up to five colors at a time. But some of the screens in Color Mixing show nine colors. This is accomplished with a display list interrupt (DLI). A DLI is a short machine language routine that, among other things, can change the contents of color registers while the computer is displaying each video frame. This technique lets you cre-

ate graphics with extra colors on various parts of the screen. (For more information, consult *De Re Atari*, published by Atari, Inc.)

The program's colored boxes are drawn in graphics mode 3 using character strings for graphics storage (lines 310–320). Here is the text equivalent of this display:

```
AAAAA CCCCC BBBBB
AAAAA CCCCC BBBBB
AAAAA CCCCC BBBBB
AAAAA CCCCC BBBBB
```

When the computer displays a string in a nontext graphics mode with the PRINT#6 statement, the letters A, B, and C show up as different-colored pixels. The letter A appears as a pixel with the color taken from color register 0; the letter B uses color register 1; and the letter C uses register 2. A SET-COLOR or POKE statement which changes the value in color register 0, for example, would change the color of the A box. This technique is used in lines 4170 and 4210–4220. Line 4270 erases the boxes by setting all their colors to black. As Color Mixing demonstrates, it's far more convenient to store these graphics in strings than to use PLOT, DRAWTO, or XIO fill statements.

## Atari Color Mixing

Please refer to "COMPUTE!'s Guide to Typing in Programs" before entering this listing.

```
N 10 DIM A$(5), B$(5), C$(5),
    D$(40), D2$(180), KDCLR
    S(4,6), BL$(35)
N 15 DIM REDS(1), GREENS(1),
    BLUE$(1), CYANS(1), MAGE
    NTAS(1), YELLOW$(1), BLA
    CK$(1)
N 20 AS="AAAAA":BS="BBBBB":
    CS="CCCCC"
N 25 BL$(1)=CHR$(32):BL$(35)
    =BL$(1):BL$(2)=BL$(1)
N 30 GRAPHICS 0:IF 7 "DNE M
    DHENT..."
N 40 FOR I=1 TO 30:READ A:D
    1$(1)=CHR$(A):NEXT I
    :D1$(31)=D1$
N 50 DATA 72,130,72,169,66,
    162,180,141,18,212,141,
    23,200
N 60 DATA 142,24,200,169,0,
    141,0,2,169,0,141,1,2
N 70 DATA 19,170,104,64
N 80 D1$(35,35)=CHR$(10):D1
    $(37,37)=CHR$(0)
N 90 D2$=D1$:D2$(61)=D1$:D2
    $(121)=D1$
N 100 ST=ADR(D1$):GOSUB 400
N 110 D1$(40,40)=CHR$(LD):D
    1$(53,53)=CHR$(HI)
N 120 ST=BT+30:GOSUB 400
N 130 D1$(10,10)=CHR$(LD):D
    1$(23,23)=CHR$(HI)
N 140 ST=ADR(D2$):GOSUB 400
```

```
U 150 D2$(140,140)=CHR$(LD):
    D2$(173,173)=CHR$(HI)
U 160 FOR I=10 TO 130 STEP
    3:ST=BT+30:GOSUB 400
U 170 D2$(1,1)=CHR$(LD):D2$
    (1+5,1+5)=CHR$(HI)
U 180 NEXT I
N 200 D2$(132,132)=CHR$(22)
U 210 OL=PEEK(560)+256*PEEK
    (561)
U 220 REBTD=240:FOR I=1 TO
    6:FOR J=1 TO 6
N 230 READ A:KOLDRS(I,J)=A:
    NEXT J:NEXT I
N 240 DATA 66,26,86,4,82,40
    26,180,116,164,4,200
N 250 DATA 86,116,164,150,1
    02,4,4,164,150,116,14
    6,180
N 260 DATA 82,4,102,146,86,
    66,40,200,4,180,66,26
N 270 REDS=CHR$(66):GREENS=
    CHR$(180):BLUE=CHR$(
    146)
N 280 CYANS=CHR$(116):MAGEN
    TAS=CHR$(86):YELLOW=
    CHR$(26):BLACKS=CHR$(
    0)
N 290 GOTO 2000
N 300 POKE B7,3:GOSUB 500
N 310 FOR I=0 TO 3:POSITION
    5,I:IF 0#A:POSITION
    16,I:IF 0#C
N 320 POSITION 31,I:IF 0#B:
    NEXT I:RETURN
N 330 POKE B7,0:GOSUB 500:R
    ETURN
N 400 HI=INT(ST/256):LO=ST-
    256*HI:RETURN
N 500 HMEM=256*HMEM+LMEM+BY
    TE
U 510 LMEM=HMEM-256*INT(HME
    M/256)
U 520 HMEM=INT(HMEM/256)
N 530 POKE BB,LMEM:POKE B9,
    HMEM:RETURN
N 600 OPEN W,1,0,"K":BET
    W1,A:CLOSE B1:RETURN
N 1000 POKE 559,0:POKE 752,
    1
N 1010 POKE OL+3,70:POKE OL
    +6,134:POKE OL+14,13
    6
N 1020 FOR I=7 TO 13:POKE O
    L+1,B:NEXT I
N 1030 ST=ADR(O1$):GOSUB 40
    0:POKE 512,LO:POKE 5
    13,HI
U 1040 LMEM=PEEK(BB):HMEM=P
    EEK(B9):POKE 710,0:P
    OKE 54206,192:RETURN
N 1200 POKE 559,0:POKE 752,
    1:REBTD=1220
U 1210 FOR I=6 TO 24:READ A
    :POKE OL+1,A:NEXT I
N 1220 DATA 134,8,8,8,136,2
    2,130,8,8,8
N 1230 DATA 136,22,130,8,8
    8,136
N 1240 POKE OL+3,70
N 1250 ST=ADR(O2$):GOSUB 40
    0:POKE 512,LO:POKE 5
    13,HI
N 1260 LMEM=PEEK(BB):HMEM=P
    EEK(B9):POKE 710,0:P
    OKE 54206,192:RETURN
N 2000 GRAPHICS 0:GOSUB 100
    0:POKE 700,146:POKE
    711,40
```

```

M2010 O1*(5,5)=GREEN*01*1
7,7)=RED*POKE 559,3
4
M2020 POKE B7,1:BYTE=0:GOSUB 500
M2030 POSITION 6,0:7 *6*:"E
(3 SPACES)RED"
M2040 POKE B7,3:BYTE=40:GOSUB 500
M2050 FOR I=2 TO 5:POSITIO
N 5,1:7 *6*:"A:POSITIO
N 17,1:7 *6*:"B:POSITIO
N 30,1:7 *6*:"C:NE
XT 1
M2060 POKE B7,0:BYTE=00:GOSUB 500
M2070 POSITION 5,0:7 "BLUE
(8 SPACES)GREEN
(9 SPACES)RED"
M2080 POSITION 2,4
M2090 ? "These are the THE
THE colors. The eye sees t
he combination"
M2100 ? "of blue, green, a
nd red as white."?
"Pairs of these prin
aries are"
M2110 ? "perceived as new
colors, the?"?
THE colors."
M2120 POSITION 5,13:7 "THE
NEXT TO GO ON, THE TO
QUIT"
M2130 GOSUB 600:IF A=155 T
HEN 2500
M2140 IF A=27 THEN GRAPHIC
S 0:END
M2150 GOTO 2130
M2500 GRAPHICS 0:GOSUB 120
0:POKE 700,146:POKE
711,40
M2530 O2*(5,5)=MAGENTA*02
(7,7)=RED*02*(65,6
5)=CYAN*
M2540 O2*(67,67)=GREEN*02
*(125,125)=RED*
M2550 O2*(95,95)=YELLOW*0
2*(127,127)=GREEN*
M2560 POKE 559,34
M2570 POKE B7,1:BYTE=0:GOSUB 500
M2580 POSITION 3,0:7 *6*:"E
(3 SPACES)RED"
M2590 BYTE=40:GOSUB 300:GOSUB 350
M2600 POSITION 5,1:7 *6*:"B
LUE(3 SPACES)+
(4 SPACES)RED
(5 SPACES)=
(4 SPACES)MAGENTA"
M2610 BYTE=120:GOSUB 300:B
YTE=40:GOSUB 350
M2620 POSITION 5,1:7 *6*:"B
LUE(3 SPACES)+
(3 SPACES)GREEN
(4 SPACES)=
(5 SPACES)CYAN"
M2630 BYTE=120:GOSUB 300:B
YTE=40:GOSUB 350
M2640 POSITION 6,1:7 *6*:"R
ED(3 SPACES)+
(3 SPACES)GREEN
(4 SPACES)=
(5 SPACES)YELLOW"
M2650 POSITION 5,1:7 *6*:"A
:POSITIO 16,1:7 *6*:"
B:POSITIO 31,1:7 *6*:"
C:NEXT 1
M2670 GOSUB 600:IF A=155 T
HEN 3000
M2680 IF A=27 THEN 2000
GOTO 2670
M2690 GRAPHICS 0:GOSUB 100
0:POKE 700,26:POKE 7
11,40
M3010 O1*(5,5)=MAGENTA*01
*(7,7)=CYAN*POKE 55
9,34
M3020 POKE B7,1:BYTE=0:GOSUB 500
M3030 POSITION 5,0:7 *6*:"E
(3 SPACES)RED"
M3040 POKE B7,3:BYTE=40:GOSUB 500
M3050 FOR I=2 TO 5:POSITIO
N 5,1:7 *6*:"A:POSITIO
N 17,1:2 *6*:"B:POSITIO
N 30,1:7 *6*:"C:NE
XT 1
M3060 POKE B7,0:BYTE=00:GOSUB 500
M3070 POSITION 4,0:7 "YELL
OW(4 SPACES)MAGENTA
(5 SPACES)CYAN"
M3080 POSITION 2,4
M3090 ? "These are the THE
THE colors. A yellow fi
iter subtracts"
M3100 ? "the blue componen
t out of white"?
"ight, only passing t
he red and"
M3110 ? "green components.
Yellow and blue"?
"are thus called THE
THE"
M3120 POSITION 5,13:7 "THE
NEXT TO GO ON, THE TO
REVIEW"
M3130 GOSUB 600:IF A=155 T
HEN 3500
M3140 IF A=27 THEN 2500
GOTO 3130
M3150 GOTO 3130
M3500 GRAPHICS 0:GOSUB 120
0:POKE 700,116:POKE
711,40
M3530 O2*(5,5)=BLUE*02*(7
,7)=MAGENTA*02*(65,
65)=GREEN*
M3540 O2*(67,67)=YELLOW*0
2*(125,125)=MAGENTA*
M3550 O2*(95,95)=YELLOW*0
2*(127,127)=RED*
M3560 POKE 559,34
M3570 POKE B7,1:BYTE=0:GOSUB 500
M3580 POSITION 1,0:7 *6*:"E
(3 SPACES)RED"
M3590 BYTE=40:GOSUB 300:GOSUB 350
M3600 POSITION 5,1:7 *6*:"C
YAN(3 SPACES)+
MAGE
NTA(4 SPACES)=
(4 SPACES)BLUE"
M3610 BYTE=120:GOSUB 300:B
YTE=40:GOSUB 350
M3620 POSITION 5,1:7 *6*:"C
YAN(3 SPACES)+
(3 SPACES)YELLOW
(4 SPACES)=
(4 SPACES)GREEN"
M3630 POKE B7,3:BYTE=120:GOSUB 500:FOR I=0 TO 3
POSITION 5,1:7 *6*:"A
:POSITIO 16,1:7 *6*:"
B:POSITIO 31,1:7 *6*:"
C:NEXT 1
M3630 BYTE=40:GOSUB 350
M3640 POSITION 4,1:7 *6*:"M
AGENTA + YELLOW
(4 SPACES)=
(5 SPACES)RED"
POSITION 5,3:7 *6*:"
NEXT TO GO ON, THE TO
REVIEW"
M3670 GOSUB 600:IF A=155 T
HEN 4000
M3680 IF A=27 THEN 3000
GOTO 3670
M4000 GRAPHICS 0:GOSUB 100
0
M4010 FOR I=0 TO 14:POKE OL+
1,6:NEXT 1:POKE OL+
9,134
M4020 FOR I=10 TO 14:POKE OL+
1,8:NEXT 1:POKE OL+
1+15,136
M4030 POKE 711,40:POKE 700
,0
M4040 O1*(5,5)=BLACK*01*(
7,7)=BLACK*
POKE 559,34
M4050 POKE B7,1:BYTE=0:GOSUB 500
M4070 POSITION 2,1:7 *6*:"E
(3 SPACES)RED"
POSITION 3,2:7 *6*:"C
YAN(3 SPACES)=
(4 SPACES)MAGENTA"
M4080 BYTE=100:GOSUB 300:B
YTE=40:GOSUB 350
M4090 POSITION 12,0:7 *6*:"
+"POSITION 24,0:7 *6*:"
M4100 POSITION 9,4:7 *6*:"1
-RED(7 SPACES)4-CYAN
"
M4110 POSITION 9,5:7 *6*:"2
-GREEN(5 SPACES)5-MAG
ENTA"
M4120 POSITION 9,6:7 *6*:"3
-BLUE(6 SPACES)6-YEL
LOW"
M4140 POSITION 2,9:7 *6*:"E
NTER A COLOR NUMBER:
"
M4145 GOSUB 600:C1=A-40
M4150 IF C1<1 OR C1>6 THEN
? CHR$(253):GOTO 4
145
M4170 ? *6*:"C1:POKE 700,KOL
ORS(C1,C1):POSITIO
N 7,0:7 *6*:"C1
M4180 POSITION 2,10:7 *6*:"
MIX 1T WITH COLOR NU
MBER: "
M4185 GOSUB 600:C2=A-40
M4190 IF C2<1 OR C2>6 THEN
? CHR$(253):GOTO 4
185
M4210 ? *6*:"C2:O1*(7,7)=CHR
$(COLOR(C2,C2)):POS
ITION 19,0:7 *6*:"C2
M4220 O1*(5,5)=CHR$(KOLORS
(C1,C2))
M4230 POSITION 6,12:7 *6*:"
NEXT TO GO MORE, THE
NEXT TO QUIT"
M4240 GOSUB 600:IF A=155 T
HEN 4270
M4250 IF A=27 THEN GRAPHIC
S 0:END
M4260 GOTO 4240
M4270 POKE 700,0:O1*(5,5)=
BLACK*O1*(7,7)=BLAC
K*
M4280 POSITION 7,0:7 *6*:"
":POSITION 19,0:7 *6*:"
M4300 FOR I=9 TO 12:POSITIO
N 2,1:7 *6*:"BL:NEXT
1:GOTO 4140

```



# Mousor:

## Escape Mode Cursor For The Apple IIc

J Blake Lambert, Assistant Editor  
Tim Victor, Editorial Programmer

*This short, fast utility makes it simple to use your Apple IIc mouse controller for editing in BASIC or the machine language monitor in escape mode.*

Despite all the improvements Apple incorporated into the IIc, the screen editing features when using BASIC or the machine language monitor are not much better than those available on the IIe. Without an editing support package, it is difficult to copy and correct program lines. And there is no way to use the mouse controller to make editing easier.

In BASIC, usually you end up making corrections by just typing the incorrect line all over again. This wastes time and effort. The alternative is to use what is called *escape mode* editing.

"Mousor" makes using escape mode easy. By rolling the mouse over an area of the desk smaller than a 3 x 5-inch index card, you can cursor (mouser) anywhere on the screen.

### How To Use Mousor

To start mousing around with Mousor, type in and save the program below. It's a BASIC loader which creates the Mousor machine language routine in memory. (Note: Save the BASIC loader on disk before running it for the first time.) When you run Mousor, it automatically checks to see if you're using DOS 3.3 or ProDOS and then adjusts itself accordingly. When the BASIC prompt reappears, you'll have a mouse-driven escape mode cursor. If you don't understand escape mode editing, see the instructions below.

The mouse is trained to work like this:

1. Click the mouse button to activate escape mode.
2. While holding the button down, roll the mouse across the desk to move the escape mode cursor.
3. Release the mouse button to exit escape mode.

Mousor locks out keypresses while it is in escape mode, so if you want to use the escape editing functions (like ESC-E to erase the end of a line), press the ESC key.

### Getting A Line Of BASIC

When you type a line of BASIC on the Apple IIc, a routine called GETLN puts the characters into a special area of memory called the input buffer. The first character on the line is stored at the start of the input buffer, and subsequent characters are added to the end of the buffer. This continues until you press the RETURN key to enter the line (with a few important exceptions). The computer clears the rest of the current screen line and stores the carriage return character into the input buffer to mark the end of the line.

When you make a mistake while entering the line, like typing the wrong character, it's easy to fix. For example, if you are entering a line such as 10 PRINT "HELLP" and notice you pressed P instead of O, you can press the left-arrow key (also called backspace) to back up and change the letter. Instead of storing the backspace in the buffer like other keypresses, GETLN treats it differently.

GETLN keeps track of the length of the input buffer by pointing to the end. When GETLN re-

ceives a backspace character, it lowers the value of the pointer by one. This removes the last character in the buffer, so all you need to do is continue typing.

If you don't notice your mistake until you've typed in several more characters, you can use the left- and right-arrow keys to make the correction. Press the backspace key until the cursor is on the letter you want to change. Type the correct letter, and then press the right-arrow key (also called retype) until the cursor returns to the end of the line.

Each time you press the retype key, the character currently under the cursor is added to the input buffer and the cursor is moved to the right. In effect, you have removed several characters from the buffer, changed the character you wanted to correct, and retrieved the rest of the characters one by one from the screen.

### Now For The Tricky Stuff

Unfortunately, you can't always catch your typing errors before you press RETURN. Often, you don't even know there's a problem in a line until you've run the program. Since the retype key allows you to pick up characters from the screen and add them to the input buffer, it would be handy if you could copy most of the bad line and type only the characters you want to change. This requires a way to move the cursor around the screen without affecting the input buffer.

Pressing the ESC key puts the IIc into escape mode. In this mode, the arrow keys move the cursor but don't change the input buffer. The IIc indicates escape mode by displaying a different cursor—an inverse plus sign. To leave escape mode, press ESC again.

Suppose you want to edit the following line in escape mode:

```
100 PRINT "THIS IS A TED"
```

If the line is not on the screen, you'll need to LIST 100. Press ESC to enter escape mode and move the cursor up to the 1. At this point, the input buffer is empty. Press ESC again and use the retype key to place the cursor on the D. This enters all but the last three characters into the input buffer. Now type the

letter S, and press retype twice, followed by RETURN. If you like, LIST 100 to verify the correction.

To edit the same line with Mouser, you would LIST the line, click the mouse button and drag the cursor to the 1, and release the mouse button. After this, follow the same editing procedure.

## Mouse Moves

You can also use escape mode to grab pieces of program lines. Mouser is especially adept at this, since movement is so easy and quick. Just keep in mind that when the mouse button is down no characters are added to the buffer.

To copy a line, first LIST it. Then enter the number for the new line you want to create, click the mouse button and drag the escape mode cursor to just beyond the original line number, and release the button. Copy the line by pressing the right-arrow key until you reach the last character, and then press RETURN.

Inserting is another useful technique. LIST the line first, then

mouser (click and drag the escape mode cursor) to the beginning of the line number. Release the mouse button and right-arrow across the line until you reach the point where you want to insert characters. Press the mouse button and mouser to a blank line on the screen, then release the button and type the insert characters. Click and drag up to the listed line again, release the button, and right-arrow to the end. After making any changes, don't forget to press the RETURN key to enter them.

## Mouser For Apple IIc

Please refer to "COMPUTE's Guide to Typing in Programs" before entering this listing.

```

10 IF PEEK (191 & 256) = 76 T
HEN GOSUB 40: GOSUB 50: GO
TO 30
20 GOSUB 50: GOSUB 40
30 FOR I = 11 TO 287: READ A:
POKE I + 768, A: NEXT I: CA
LL 760: END
40 FOR I = 0 TO 10: READ A: P
OKE I + 768, A: NEXT I: RETU
RN
50 FOR I = 0 TO 10: READ A: N
EXT I: RETURN
60 DATA 216, 169, 67, 141, 50, 190
70 DATA 169, 3, 141, 51, 190

```

```

80 DATA 169, 67, 133, 56, 169, 3
90 DATA 133, 57, 32, 234, 3
100 DATA 120, 162, 196, 160, 64, 3
2
110 DATA 28, 196, 169, 0, 141, 120
120 DATA 4, 141, 120, 5, 141, 240
130 DATA 5, 169, 0, 141, 240, 4
140 DATA 169, 0, 162, 196, 160, 64
150 DATA 32, 176, 196, 169, 1, 162
160 DATA 196, 32, 176, 196, 162, 1
96
170 DATA 32, 132, 196, 169, 1, 162
180 DATA 196, 160, 64, 32, 61, 196
190 DATA 00, 96, 145, 40, 32, 76
200 DATA 284, 44, 99, 192, 16, 8
210 DATA 32, 112, 284, 16, 246, 76
220 DATA 37, 253, 218, 90, 72, 169
230 DATA 4, 141, 124, 4, 141, 252
240 DATA 4, 32, 187, 3, 32, 195
250 DATA 3, 44, 99, 192, 16, 9
260 DATA 184, 32, 179, 195, 122, 2
50
270 DATA 76, 69, 3, 173, 124, 4
280 DATA 240, 7, 201, 8, 144, 25
290 DATA 162, 156, 44, 162, 136, 1
04
300 DATA 32, 179, 195, 138, 32, 00
310 DATA 285, 32, 195, 3, 72, 169
320 DATA 4, 141, 124, 4, 32, 187
330 DATA 3, 173, 252, 4, 240, 7
340 DATA 281, 8, 144, 25, 162, 138
350 DATA 44, 162, 159, 164, 32, 17
9
360 DATA 195, 138, 32, 00, 285, 32
370 DATA 195, 3, 72, 169, 4, 141
380 DATA 252, 4, 32, 187, 3, 76
390 DATA 182, 3, 162, 196, 160, 64
400 DATA 32, 187, 196, 96, 32, 29
410 DATA 284, 72, 41, 120, 73, 171
420 DATA 32, 179, 195, 164, 96

```

# Commodore 64 Headliner

Robert F. Lambiose

Create attention-getting headlines and titles with this oversized alphabet for the Commodore 64.

Nearly every program uses titles or headlines of some sort, and you ordinarily want titles to look as impressive as possible. But the standard Commodore character set doesn't permit much variety. You can use different character colors or print in reverse video, but the letters are still pretty small. "Commodore 64 Headliner" lets you create truly striking titles and headlines with an alphabet that's four times bigger than normal.

Enter and save Headliner from the listing below, then run it. After a short pause to form the new char-

acters, the program prints the alphabet in giant, quadruple-size characters. The letters can be any color, and the standard-size alphabet is available, too. The only thing you give up are reverse video characters, since Headliner redefines them as large characters.

## Using Headliner

Headliner is easy to incorporate in your own programs. The first step is to include lines 100-350 (they can be renumbered, of course) to create the new character set. The program begins storing the new character definitions at memory location 12288. The statement POKE 53272,29 (see line 240) tells the computer to look at this memory area for character definition data. Use POKE 53272,21 to switch back

to normal characters.

After defining the new characters, Headliner prints the expanded alphabet (lines 400-405) and a title (lines 410-430). Whenever a string of large characters is to be printed, the characters are defined as a string (X\$). Then two important variables (SL and CC) are defined. Finally, the statement GOSUB 500 calls the subroutine that puts the big characters on the screen. The subroutine analyzes each character in X\$; if it is not a space character, its pattern is POKED into screen memory.

The variable SL sets the position of the large characters on your screen. The 64's screen is divided into 25 rows and 40 columns, giving a total of 1,000 different loca-

tions. Each screen memory location has a different address, and they are numbered in order, beginning at the upper-left corner of the screen. The upper-left screen position is location 1024; the next location to the right is 1025, and so on. Color memory is a second 1,000-byte memory area that corresponds to screen memory. By POKEing the right number into color memory, you can control the color of any screen memory location. Your *Commodore 64 User's Guide* has maps that show the numbers for every screen memory and color memory location, as well as a list of all the color numbers.

To place large characters on the screen, find the location you want using the screen memory map in your user's guide, then set SL to that value. The upper-left corner of the first large character appears at the location defined by SL, and the others follow in order. For example, to start printing large characters at the upper-left corner of the screen, use the statement SL=1024.



Note that the title above the alphabet uses a large character to begin a line of standard characters, somewhat like a super-capital letter. Line 420 of the program sets the computer to start printing again at the next location after the last expanded character.

### Pick Your Colors

The variable CC sets the color of the large characters using the color numbers listed in your user's guide. Line 410 of Headliner uses the statement CC=3 to print in cyan. Use the statement CC=1 to print in white, and so on.

There may be times when you want to print large characters in the current character color. Line 400 of Headliner does this with the state-

ment CC=PEEK(646). Location 646 always contains the current color number for PRINTING characters.

The 64 actually has two alternate character sets: One is used in uppercase/graphics mode, and the other is used in lowercase/uppercase mode. You can switch from one mode to another by pressing SHIFT-COMMODORE. Since Headliner works only in uppercase/graphics mode, you should disable the SHIFT-COMMODORE key combination to prevent the user from accidentally destroying the display. To do this, insert PRINT CHR\$(8) at the beginning of your program. When the program ends, type PRINT CHR\$(9) to restore things to normal.

### Building Giant Characters

Each large character is actually four redefined characters placed together. (To see this more clearly, type POKE 53272,21 and press RETURN after the program has run.) The standard Commodore 64 character set contains 256 characters, numbered from 0-255. Characters 0-127 are the "normal" characters and characters 128-255 are the same characters in reverse video. Since each character definition takes eight bytes, a full set of character definitions requires 2,048 (8\*256) bytes.

The first step in redefining characters is to copy the standard character set from ROM (Read Only Memory) into RAM (Random Access Memory) where it can be altered. The program does this in lines 200-240. The new character set begins at location 12288. Since we only want characters 0-127 from the standard set, only those character definitions are copied.

Next the program POKEs the expanded character definitions into the memory area that would otherwise store reverse video character data (see lines 300-350). The bit pattern of each standard character is mapped into a four-character-sized memory area, using conversion values stored in the T() array.

Since each large character definition takes four times the memory of a standard definition, we have room for a maximum of 32 (128/4) expanded definitions. That's

enough for 26 letters, but not enough space to hold ten numeral definitions, too. However, you could squeeze in six more characters—perhaps punctuation or other symbols.

By sacrificing all the standard characters, you can get as many as 64 large characters—but remember to define a space character so you can still clear the screen. The *Commodore 64 Programmer's Reference Guide* contains much more information about using redefined characters.

### Headliner

Please refer to "COMPUTE's Guide to Typing in Programs" before entering this listing.

```

100 DIMT(15):POKE53281,0:POKE5
3280,0:PRINTCHR$(0):G=5427
2:          :ren 24
110 POKE646,1:FORJ=0TO15:READT
(J):NEXT:          :ren 93
120 DATA 0,3,12,15,48,51,60,63
,192,195,204,207,240,243,2
52,255:          :ren 220
200 PRINTCHR$(147)TAB(125)*DOW
NLOADING THE CHARACTER SET
":G=53248:GN=12288:ren 110
220 POKE 56333,127:POKE51,1:FO
R G=0TO1023:POKEGN+G,PEEK(
G+G):NEXT:          :ren 89
240 POKE1,55:POKE56333,129:POK
E53272,29:          :ren 196
300 PRINTCHR$(147)TAB(125)*FOR
MING THE LARGE CHARACTERS":
:POKE13312,0:          :ren 15
320 FORR=0TO128STEP8:R1=12288+
R:R2=13312+4*R:          :ren 19
330 FORI=0TO4STEP4:FORK=0TO3:
=PEEK(R1+K+I):B=B+2*(K+I)
+1:          :ren 233
340 XI=T((JAND240)/16):X2=T(JA
ND15):          :ren 197
350 POKE,X1:POKEK+1,X1:POKEK+
16,X2:POKEK+17,X2:NEXTK,I,
R:          :ren 72
360 PRINTCHR$(147):          :ren 20
400 X$="ABCDEFGHIJKLM":SL=1270
:CC=PEEK(646):GOSUB 500
:          :ren 110
405 X$="NOPQRSTUVWXYZ":SL=1350
:CC=PEEK(646):GOSUB500
:          :ren 27
410 X$="H":SL=1150:CC=3:GOSUB
(SPACE)500:          :ren 62
420 NS=SL+40:POKE210,INT(NS/25
6):POKE209,NS AND 255
:          :ren 179
430 POKE646,7:PRINT"HEADLINER":
FORJ=1TO10:PRINTNEXT:END
:          :ren 70
500 FORP=1TOLEN(X$):L=(ASC(MID
$(X$,P,1))-64)*4+120:PL=P-
THEN550:          :ren 111
530 POKESL+G,CC:POKESL+1+G,CC:
POKESL+4+G+G,CC:POKESL+4+G
,CC:          :ren 137
540 POKESL,L:POKESL+1,L+2:POKE
SL+40,L+1:POKESL+41,L+3
:          :ren 241
550 SL=SL+2:NEXT:RETURN:ren 5

```

# Using The Commodore USR Function

Keith R. Bergerstock

*The USR function provides a convenient way for BASIC programs to call machine language subroutines—and it's more versatile than the SYS statement. Although this article is oriented toward the Commodore 64, the general principles apply to all Commodore computers. A demonstration program shows how to add five new functions to Commodore 64 BASIC.*

It's often overlooked, but the USR function is a powerful and convenient tool for accessing machine language (ML) routines from BASIC. In its simplest form, USR works just like the more familiar SYS command. SYS makes the computer halt BASIC program execution and jump to an ML routine at a specified address. When the ML routine is done, BASIC resumes what it was doing. SYS lets you jump anywhere in the computer's memory, to a system routine stored in Read Only Memory (ROM) or a user-written ML program stored in free memory.

To see an example on the Commodore 64, move the cursor to a blank line somewhere near the middle of the screen, type SYS 59626, and press RETURN. SYS 59626 jumps to the computer's ROM scrolling routine: The screen scrolls up and the blinking cursor reappears.

Although USR requires a little preparation, it's much easier to use after the preparation is done. Let's call the same scrolling routine with USR. Type the following line in direct mode (without a line number) and press RETURN.

```
POKE 785,234: POKE 786,232:  
A=USR(0)
```

The screen scrolls upward, just as it did when you typed SYS

59626. The POKES set up the routine's address for USR. This method looks cumbersome, but the POKES are needed only once. Afterward you can call the scrolling routine whenever you like, just by entering A=USR(0). Program 1 below contains a formula that automatically performs the correct POKES to prepare any address for USR.

Like PEEK and other BASIC functions, USR must be followed by a value in parentheses. However, in the simplest case (when you just want to jump to an ML routine), the value and the preceding variable name are both irrelevant: You get the same result with A=USR(XYZ) or GG=USR(12345678). You can even use PRINT USR(X), though that usually prints something on the screen.

## Parameter Passing

The real value of USR lies in its ability to pass parameters (values) back and forth between BASIC and machine language. To see how this works, type in and save Programs 1 and 2 below. Then run Program 1; it puts a short, multipurpose ML program in memory and sets up the USR address vector (a pair of memory locations that point to the ML routine).

The variable SA in line 10 defines the starting address of the ML routine. This ML program is relocatable, so you can put it elsewhere if you like. For instance, to put the routine at 49152, change line 10 so that SA=49152 and re-run Program 1.

The rest of line 10 converts the address into low byte/high byte format for the USR vector. Since 255 is the largest number any single memory location can hold, the

computer must use two adjacent locations to store addresses like 59626. Program 1 stores the high byte of the address in the variable HI% and the low byte in LO.

Line 20 POKES LO and HI% into vector locations 785 and 786. You must always put the target address in these locations before using USR. The rest of Program 1 POKES the ML into the computer's memory. To use this technique in your own programs, just duplicate the method shown in Program 1.

If you get an ?ILLEGAL QUANTITY error message when experimenting with USR, it probably means that you forgot to put a vector address in 785-786. When you turn the computer on, the vector in 785-786 points to 45640, the BASIC routine that prints that message.

USR works virtually the same on all Commodore computers; the only difference is the location of the USR vector. You'll find it at locations 1-2 on the VIC-20 and 812-813 on the Commodore Plus/4 and 16. The other vectors mentioned below also are located in different places on various machines.

## Using USR

Program 1 provides five handy functions which you select by inserting a number from 0-4 in the parentheses after USR. For an illustration, plug a joystick in port 2, then load and run Program 2 after running Program 1 to install the ML. As you move the joystick, the program prints the joystick directions on the screen. To exit the program, press the fire button.

Line 50 of Program 2 does the important work. The statement

JV=USR(3) calls the ML routine and selects function 3 (read joystick). Each time the ML routine performs this function, it gives the variable JV a numeric value representing the joystick position. JV is 0 when the joystick is centered, 9 when the fire button is pressed, and so on.

Note that Program 2 passes information in both directions. The value in parentheses—USR(3)—sends information to the ML routine (telling it to perform function 3). And the ML routine passes other information back in the form of a variable (JV).

The other four functions work in similar fashion. Function 4 returns the size of the BASIC program currently in memory. Whenever you want to know your program size, use the statements A=USR(4): PRINT A. Function 2 changes the screen background and border colors. To activate this function, use A=USR(C\*256+2), replacing C with the number of the screen color you want (your 64 User's Guide lists the color numbers).

Function 1 reads the Y and N keys, returning the value of 1 when Y is pressed, and 2 when N is pressed. The ML routine waits until you press Y or N, ignoring all other keys. This function is useful in the common case where a program asks the user a Yes/No question. Combining USR with ON-GOTO or ON-GOSUB is a very efficient technique. For instance, type in and run the following program (make sure the ML routine is in memory):

```
10 PRINT"ENTER Y/N:ON USR(1)
   GOSUB 100,200
20 GOTO 10
100 PRINT"YES":RETURN
200 PRINT"NO":RETURN
```

Function 0 is similar to Function 1, but reads the eight special function keys. Enter and run this program to see how it works:

```
10 A=USR(0)
20 PRINT A:GOTO 10
```

In this case USR returns a number from 1-8 in the variable A. (But note that the numbers returned don't correspond directly to the function key numbers. Keys f1, f3, f5, and f7 return values 1-4 respectively, while the shifted keys, f2, f4, f6, and f8, return values 5-8.) In your own pro-

grams, of course, you can use any variable name you like; this function could also be used with ON-GOTO or ON-GOSUB to select as many as eight different options.

## The Facts About FAC1

If you simply want to use the new functions provided by Program 1, you needn't read any further. If you're ready to write your own ML routines for USR, here are a few additional tips. First, when you pass a value from BASIC, the value is converted into a different number format (floating point) and placed in the computer's floating point accumulator (locations 98-101).

The floating point accumulator—usually called FAC1 to distinguish it from the secondary accumulator—is a special number-processing area used internally for many purposes. Since floating point numbers are quite difficult to handle, it's helpful to convert the floating point value into an integer (whole number) before using it in your ML routine. Fortunately, the 64 has built-in routines to convert floating point numbers to integers and vice versa. These routines can be accessed directly with JSR, or indirectly through the vectors in locations 3-4 and 5-6.

The routine at location 45482 (\$B1AA) converts a floating point number in FAC1 to an integer. Use this routine when passing a value from BASIC to ML. At the point where you want to retrieve the passed value, use JSR \$B1AA to do the conversion. The computer returns the low byte of the integer in the Y register and the high byte in the A register. If you'd rather use the vector, use LDA #\$4C: STA \$02: JSR \$0002.

Passing a value from ML back to BASIC often requires the opposite conversion. The routine at 45969 (\$B391) converts an integer to floating point format and stores the result in FAC1. At the point where you want to return to BASIC, load the low byte of your integer value in the Y register and the high byte in A. Then call the integer-to-floating point conversion routine with JSR \$B391: RTS (you can also compress these two instructions into JMP \$B391). The value is converted and stored in FAC1, and

RTS returns you to BASIC. If you prefer to use the vector, JMP (\$0005) accomplishes the same thing.

Like other vectors, the vectors at 3-4 and 5-6 will presumably be safe to use even if the actual ROM addresses of the routines change after a ROM update. However, there's one danger in using them. Since BASIC never uses locations 2-6, many ML programmers use them as free zero page space. If your routine jumps through these vectors after some other ML program overwrites them, it may send the computer into never-never land.

## Program 1: USR Loader For Commodore 64

```
10 SA=53008:HI%SA=SA/256:LO=SA-HI*256          :rem 23
20 POKE785,LO:POKE786,HI:CK=0      :rem 33
30 READQ:IFQ>-1THENPOKE$A,Q:SA=SA+1:CK=CK+Q/6707030 :rem 213
40 IFCK=14485THENPRINT"OK":NEW      :rem 138
50 PRINT"ERROR IN DATA STATEMENT NTS" :rem 121
60 DATA 169,0,133,196,169,76,1     :rem 254
   DATA 33,32,2,0,132,2             :rem 254
61 DATA 192,0,248,19,192,1,248     :rem 197
   DATA 35,192,2,248,50             :rem 197
62 DATA 192,3,248,71,192,6,248     :rem 43
   DATA 54,198,0,3,32,228           :rem 43
63 DATA 255,281,133,144,249,28     :rem 218
   DATA 1,141,176,245,56            :rem 218
64 DATA 233,132,168,169,0,188      :rem 51
   DATA 5,0,32,228,255,281          :rem 51
65 DATA 89,248,8,281,78,288,24     :rem 177
   DATA 5,168,2,288,236             :rem 177
66 DATA 168,1,288,232,141,32,2     :rem 54
   DATA 88,141,3,208                :rem 54
67 DATA 169,0,168,188,5,0,56,1     :rem 235
   DATA 65,45,229,43,168           :rem 235
68 DATA 165,46,229,44,188,5,0,     :rem 162
   DATA 173,0,228,73,31             :rem 162
69 DATA 41,31,168,201,3,144,12     :rem 204
   DATA 136,281,8,144,7            :rem 204
70 DATA 136,281,16,144,2,168,9     :rem 236
   DATA 169,0,188,5,0,-1           :rem 236
```

## Program 2: Joystick Demo For Commodore 64

```
10 DATA NONE,UP,DOWN,LEFT,UP/L
   EFT :rem 98
20 DATA DOWN/LEFT,RIGHT,UP/RIG
   HT :rem 54
30 DATA DOWN/RIGHT,FIRE BUTTON
   :rem 144
40 PRINTCHR$(147):POB%←OT09:IRE
   ADAS(J):SP$←SP$+CHR$(32):NE
   XT :rem 73
50 JV=USR(3):PRINTCHR$(19)JV,A
   ($JV)SP$ :rem 44
60 IFJV=9THENE=E+1 :rem 93
70 IF%←20THENE=50 :rem 109
```

# Sound And Music On The Commodore 128 Part 1

Philip I. Nelson, Assistant Editor

*The Commodore 128's advanced BASIC makes it easy and fun to create music or sound effects. Part 1 of this two-part series shows how to use the VOL, TEMPO, and ENVELOPE statements. Part 2 explores the FILTER, SOUND, and PLAY commands and includes three short tutorial programs.*

If you've heard much about the new Commodore 128, you probably know that it contains a very powerful music maker: the SID (Sound Interface Device) chip, exactly as found in the Commodore 64 and still the best sound chip in any personal computer. The SID chip provides three independent voices (tone generators) for playing up to three notes at once, and four different waveforms to simulate virtually any sound.

Although both computers use the SID chip, the comparison ends there. Since Commodore 64 BASIC has no sound commands, even simple 64 sound effects require several POKE statements. The 128's BASIC eliminates the POKes by adding six new music and sound commands: PLAY, SOUND, VOL, TEMPO, ENVELOPE, and FILTER.

## Simplicity And Power

The PLAY command is both powerful and easy to use. If you have access to a 128, type in and run the

following one-line program. (The spaces make the statement more readable, but are not necessary.)

```
100 PLAY "C D E F G F E D C"
```

The 128 plays nine notes, going up the scale and down again. It would take a lot more work to play the same nine notes on the 64—you'd need at least three preliminary POKes (to set the volume and sound envelope), plus four POKes for each note (one to turn on the voice, two to set the pitch, and one to turn off the voice).

Interestingly, you can control the SID chip in 128 mode with the same POKes as on the 64. That's usually a waste of time, since the 128's BASIC commands are more convenient than POKes. However, 128 BASIC has certain limitations (SOUND statements can't use ring modulation or synchronization, for example). If you already know sound programming on the 64, you may still find uses for old-fashioned 64 programming techniques.

The PLAY command is so versatile that it's almost a mini-language in itself. In addition to playing notes, you can insert rests, change octaves, choose any of ten different instrument voices, use filtering, and even play multivoice music. This month we'll stick to simple PLAY statements and examine the VOL, TEMPO, and ENVELOPE commands in detail. In Part 2, we'll look at the FILTER and SOUND commands and more advanced uses of PLAY.

## VOL Means Volume

The 128's VOL command affects all three voices at once and accepts values from 0 (silence) to 15 (maximum). Add the following line to the example program and run it again:

```
10 VOL 15
```

Since the song plays at the same volume, it seems VOL had no effect. In fact, VOL just duplicated the default volume setting that PLAY uses when no volume is specified. When you turn on the 128, it establishes several music and sound settings (parameters) in advance. For instance, the PLAY statement above plays the notes at maximum volume with a sound envelope and waveform that simulate a piano. Other default sound parameters, too, remain in effect until you change them.

In many cases you can set the volume at the beginning of a program and leave it alone. However, gradual changes in volume can add to the dynamics of a song. Since drastic volume changes make the SID chip "pop," don't use VOL to turn individual notes on and off. (To hear the pop, turn up the volume on your monitor or TV set, enter the following line without a line number, and press RETURN: VOL 15:VOL 0:VOL 15.)

Unlike PLAY statements, SOUND statements (to be discussed in Part 2) default to a volume of 0. Before using SOUND you must always use VOL to set the volume to some nonzero value.

## TEMPO

TEMPO is another command that affects all voices equally, setting the speed at which a song plays. TEMPO is followed by one number in the range 0-255. The default tempo setting is 15, a pedestrian speed. Add the following line to the example program and run it again:

```
20 TEMPO 50
```

At a tempo of 50, the song plays much faster. Try several different TEMPO values in line 20. As you'll find, the highest tempos are exceedingly fast—too speedy for playing whole songs, but handy for simulating trills and grace notes. Change the TEMPO value back to 15 when you're done experimenting with line 20.

Don't confuse tempo—the overall speed of the music—with the individual *duration* of each note (quarter note, sixteenth, etc.). In conventional music a quarter note lasts one "beat," an eighth note lasts one-half beat, and so on. Tempo defines how many beats are played in a minute. At faster tempos every note plays faster, but quarter notes still last twice as long as eighth notes. The default note duration for PLAY is a quarter note.

## A Built-In Orchestra

The ENVELOPE command is more versatile than VOL or TEMPO. It is used to create customized instrument sounds for your songs. ENVELOPE takes the following general form:

```
ENVELOPE i, a, d, s, r, w, p
```

In the above example, *i* stands for the instrument number, *a* for the attack rate, *d* for decay rate, *s* for sustain rate, *r* for release rate, *w* for waveform, and *p* for pulsewidth. Naturally, in a program these letters are replaced with appropriate numbers.

The first number in an ENVELOPE statement chooses one of the 128's instrument voices. There are ten predefined instruments, numbered 0-9 as shown here:

Instrument	ENVELOPE #
Piano	0
Accordion	1
Calliope	2
Drum	3
Flute	4
Guitar	5

Harpichord	6
Organ	7
Trumpet	8
Xylophone	9

Since PLAY commands use the same instrument numbers, you'll want to become familiar with this list. To pick an instrument within PLAY, add a *T* (for *tune*) followed by the desired instrument number. For instance, PLAY "T5 C D T3 E F" selects instrument 5 (guitar) and plays notes C and D, then selects instrument 3 (drum) and plays notes E and F. The same numbering scheme identifies customized instruments, as you'll see in a moment. The default instrument for PLAY statements is instrument 0 (piano); if you don't specify an instrument, PLAY always produces a piano sound.

## Sound Envelopes

To create new instrument sounds, you'll need to learn about sound envelopes and waveforms. Every natural sound has a distinctive *envelope* or sound pattern. Consider the difference between a snare drum and a violin. Drum sounds begin and end very sharply. The drumhead starts vibrating the instant you strike it, and fades quickly. Violin sounds start out more softly, as the string gradually picks up vibrations from the bow, and fade softly as the vibration dissipates.

The 128 defines different sound envelopes in terms of four values: attack, decay, sustain, and release (ADSR). The *attack* value defines how quickly the sound rises from silence to its peak volume. *Decay* defines how quickly the sound fades from peak volume to the volume at which it will be sustained (held). *Sustain* sets the volume level for the sound's main duration. *Release* defines how quickly the sound fades from its sustained volume back to silence again. Figure 1 illustrates a typical sound envelope.

In ENVELOPE statements, the four numbers after the instrument number define the ADSR envelope. ADSR numbers can range from 0-15.

## Waveforms

ENVELOPE also lets you pick different *waveforms*. Each of the SID

chip's three voices can produce four different waveforms, diagrammed in Figure 2. The *triangle* waveform (used for the flute, instrument 4) is soft and rich. The *sawtooth* wave (used for the guitar, instrument 5) creates a louder, harsher sound.

The *pulse* waveform (used for the organ, instrument 7) is the most versatile of all. It's louder than the triangle wave and can be adjusted to make sounds that are rich and full or thin and faint. The *noise* waveform (used for the drum, instrument 3) is a random mish-mash of frequencies that make a hissing or rushing sound. ENVELOPE uses the following waveform numbers:

Number	Waveform
0	Triangle
1	Sawtooth
2	Pulse
3	Noise
4	Ring Modulation

Ring modulation is a special effect, different from the other waveforms. The SID chip creates ring modulation by combining the frequencies of two voices into one complex sound. Note that ENVELOPE cannot use *synchronization*, another SID effect familiar to 64 programmers.

Finally, ENVELOPE lets you choose different *pulsewidth* values for the pulse waveform (2). The pulsewidth number can range from 0-4095. Look again at the pulse wave diagram in Figure 2. The top portion of each wave is wider than the bottom portion. The pulsewidth value defines the ratio between these two parts of the wave. Medium pulsewidth values (roughly from 1000-3000) produce fairly symmetrical waves and full, solid tones. Very small or very large pulsewidth values produce asymmetrical waves and thin, hollow tones.

## ENVELOPE With PLAY

To see what ENVELOPE can do, add line 30 to the example program and insert T1 in line 100 as shown below:

```
30 ENVELOPE 1, 7, 0, 0, 0, 2, 2000  
100 PLAY "T1 C D E F G F E D C"
```

Run the program again and notice how different the new instrument sounds. Line 30 selects instrument 1, sets attack at 7, decay, sustain, and release at 0, waveform

at 2 (pulse), and pulsewidth at 2000.

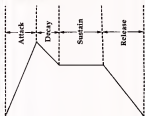
The T1 in line 100 might seem redundant at first: If ENVELOPE selects instrument 1, why specify instrument 1 again in the PLAY statement? This is necessary because of the default system. Until you specify otherwise in a PLAY statement, PLAY always uses instrument 0, the piano. Thus, whenever you define a new instrument with ENVELOPE, you must use the same instrument number after T in the appropriate PLAY statement. If you forget, PLAY ignores the ENVELOPE statement and uses instrument 0 or whatever instrument you last selected with T.

Redefining an instrument with ENVELOPE replaces the predefined instrument of that number. Thus, you can never have more than ten instruments at once. However, new instruments can be introduced at any time with new ENVELOPE statements.

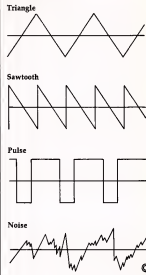
ENVELOPE can be tricky to handle, since it gives you total control over the ADSR envelope and

must be properly integrated with other sound commands. For instance, an envelope that sounds fine at slow tempos may be unsuitable at faster tempos. Don't be discouraged if your first experiments sometimes fail. Remember, ENVELOPE is necessary only for customized instrument sounds. If you're happy with the predefined instruments, just use T in a PLAY statement to choose the one you want.

**Figure 1. Typical Sound Envelope**



**Figure 2. Commodore 128 Waveforms**



## Colorful Text For IBM Graphics

Peter F. Nicholson, Jr.

IBM personal computers provide a wealth of graphics modes. Here's a method of printing text on graphics screens with different foreground colors to brighten up your screen displays. The technique works on the IBM PCjr with Cartridge BASIC or the PC with BASICA and color/graphics adapter card.

The IBM PCjr and the PC with a color/graphics adapter both have the ability to print different-colored foreground characters on the same colored background. Medium-resolution graphics mode gives you a choice of 16 different background colors and two different foreground color palettes (red/brown/green or cyan/magenta/white). According

to page I-9 of the BASIC reference manual for the PC, the following statement changes character colors (substitute 1, 2, or 3 for color number):

**DEF SEG:POKE &H4E,color number**

Although this changes the character color, it also replaces the background color with color 0



(black). It's not mentioned in the manual, but there is a way to change the foreground color without losing the background color. Simply add 128 to the color number in the above statement. This performs a bit manipulation called an *exclusive OR (XOR)* of the color value, allowing you to print any foreground color on the background.

To see a demonstration, type in and save the program below, then run it. The program illustrates the difference between normal and XOR printing and lets you experiment with many different background and foreground color combinations.

### Character Colors For XOR Printing

Color Where Character Will Be XOR PRINTed	Resulting Character Color			
	&H4E=	&H4E=	&H4E=	
	129	130	131	
0	1	2	3	
1	0	3	2	
2	3	0	1	
3	2	1	0	

Remember to restore the value in memory location &H4E to either 1, 2, or 3 when you're editing. Otherwise anything you type is XORed with whatever is on the screen. The easiest way to avoid this problem is to clear the screen (press CTRL-HOME) and press function key 10 to edit in SCREEN 0. The table shows the values you can POKE into &H4E to generate various color combinations.

### XOR Printing

```

# 100 SCREEN 1:KEY OFF:PAL=0:BA
CK=0:GOSUB 1500:COLOR BAC
K,PAL
# 110 CLS:COL=3:X0=40:Y0=20:GOS
UB 1000:COL=0:GOSUB 1000
# 120 COL=3:X0=40:Y0=116:GOSUB
1000:COL=0:GOSUB 1000
# 130 GOSUB 2000:POKE &H4E,1:L
O CATE 3,0:PRINT "&H4E=1"
# 140 FOR I=5 TO 9 STEP 2:LOCAT
E 1,0:PRINT PAL*(PAL,1):
NEXT I
# 150 POKE &H4E,2:LOCATE 3,18:P
RINT "&H4E=2"
# 160 FOR I=5 TO 9 STEP 2:LOCAT
E 1,18:PRINT PAL*(PAL,2):
NEXT I
# 170 POKE &H4E,3:LOCATE 3,28:P
RINT "&H4E=3"

```

```

# 180 FOR I=5 TO 9 STEP 2:LOCAT
E 1,28:PRINT PAL*(PAL,3):
NEXT I
# 190 POKE &H4E,1:LOCATE 14,7:P
RINT "&H4E=129":POKE &H4E
,129
# 200 FOR I=16 TO 20 STEP 2:LOC
ATE 1,0:PRINT COL*(FIX((
I-16)/2)):NEXT I
# 210 POKE &H4E,2:LOCATE 14,17:
PRINT "&H4E=130":POKE &H4E
,130
# 220 FOR I=16 TO 20 STEP 2:LOC
ATE 1,18:PRINT COL*(FIX(
I-16)/2)):NEXT I
# 230 POKE &H4E,3:LOCATE 14,27:
PRINT "&H4E=131":POKE &H4E
,131
# 240 FOR I=16 TO 20 STEP 2:LOC
ATE 1,28:PRINT COL*(FIX(
I-16)/2)):NEXT I
# 250 POKE &H4E,3:LOCATE 1,1:PR
INT "Normal Printing":
# 260 LOCATE 12,1:PRINT "XOR Pr
inting":
# 270 LOCATE 22,1:PRINT STRING$
(40,CHR$(32)):
# 280 LOCATE 22,1:PRINT "Backgr
ound ";BACK$(BACK):LOCAT
E 22,25:PRINT "Palette ";
PAL
# 290 LOCATE 23,1:PRINT "Press
Q To Quit"+STRING$(24,CHR
$(32)):
# 300 LOCATE 24,1:PRINT "Press
Esc To Change Palette":
# 310 LOCATE 25,1:PRINT "Press
Space Bar To Change Backg
round":
# 320 KB$=INKEY$:IF KB$="" THEN
330 ELSE 320
# 330 KB$=INKEY$:IF KB$="" THEN
330
# 340 IF KB$="q" OR KB$="o" THE
N CLS:END
# 350 IF ASC(KB$)=32 THEN 300 E
LSE IF ASC(KB$)>>27 THEN
330
# 360 IF PAL=1 THEN PAL=0 ELSE
PAL=1
# 370 COLOR BACK,PAL:GOTO 130
# 380 LOCATE 25,1:PRINT STRING$
(30,CHR$(32)):
# 390 LOCATE 24,1:PRINT STRING$
(30,CHR$(32)):
# 400 LOCATE 23,1:INPUT "Enter
Color Number (0-15) ":BAC
K
# 410 COLOR BACK,PAL:GOTO 130
# 1000 PRESET (X0,Y0)
# 1010 FOR I=1 TO 3
# 1020 LINE STEP(0,0)-STEP(240,
16),COL,B
# 1030 PRESET STEP(-240,0)
# 1040 NEXT I
# 1050 PRESET (X0,Y0)
# 1060 FOR I=1 TO 3
# 1070 LINE STEP(0,0)-STEP(80,4
0),COL,B
# 1080 PRESET STEP(0,-40)
# 1090 NEXT I:IF COL=0 THEN GOT
O 1130
# 1100 FOR I=1 TO 3:FOR J=1 TO
3
# 1110 PRINT(X0+20+80*(I-1),Y0+
8+16*(J-1)),J,3
# 1120 NEXT J:NEXT I
# 1130 RETURN
# 1500 DIM PAL$(1,3)
# 1510 FOR I=0 TO 3:FOR J=1 TO
3:READ PAL$(I,J):NEXT J
# 1520 DATA "GREEN","RED","BRO
N",

```

```

N","CYAN","MAGENTA","WHI
TE"
# 1530 DIM BACK$(15)
# 1540 FOR I=0 TO 15:READ BACK$
(I):NEXT I
# 1550 DATA "BLACK","BLUE","GRE
EN","CYAN","RED","MAGEN
T","BROWN"
# 1560 DATA "WHITE","GRAY","L B
LUE","L GREEN","L CYAN",
"L RED","L MAGN"
# 1570 DATA "YELLOW","HI WHITE"
# 1580 BLOCK$=STRING$(8,CHR$(25
5)):B$=PEEK(IVARPTR(BLOCK
$)+1):B$=PEEK(IVARPTR(BLO
CK$)+2)
# 1590 DEF SEG=0:POKE 124,AK:PO
KE 125,B$:POKE 126,PEEK(
&H510):POKE 127,PEEK(&H5
11)
# 1600 DEF SEG:RETURN
# 2000 COL1$(0)=BACK$(BACK):COL
1$(1)=PAL*(PAL,3):COL1$(
2)=PAL*(PAL,2)
# 2010 COL2$(0)=PAL*(PAL,3):COL
2$(1)=BACK$(BACK):COL2$(
2)=PAL*(PAL,1)
# 2020 COL3$(0)=PAL*(PAL,2):COL
3$(1)=PAL*(PAL,1):COL3$(
2)=BACK$(BACK)
# 2030 DEF SEG:POKE &H4E,1
# 2040 FOR I=0 TO 20 STEP 10:L
O CATE 5,1:PRINT STRING$(8
,CHR$(128)):NEXT I
# 2050 FOR I=0 TO 20 STEP 10:L
O CATE 16,1:PRINT STRING$(
8,CHR$(128)):NEXT I
# 2060 POKE &H4E,2
# 2070 FOR I=0 TO 20 STEP 10:L
O CATE 7,1:PRINT STRING$(8
,CHR$(128)):NEXT I
# 2080 FOR I=0 TO 20 STEP 10:L
O CATE 18,1:PRINT STRING$(
8,CHR$(128)):NEXT I
# 2090 POKE &H4E,3
# 2100 FOR I=0 TO 20 STEP 10:L
O CATE 9,1:PRINT STRING$(8
,CHR$(128)):NEXT I
# 2110 FOR I=0 TO 20 STEP 10:L
O CATE 20,1:PRINT STRING$(
8,CHR$(128)):NEXT I
# 2120 RETURN

```

Copies of articles from this publication are now available from the UMI Article Clearinghouse.

Net I would like to know more about UMI Article Clearinghouse: I am interested in electronic ordering through the following services:

- ☐ OALOC/Enduser ☐ ITT Telecom  
☐ OnType ☐ COLC BIL Subscription

☐ Other (please specify):

☐ I am interested in sending the order by mail

☐ Please send me your current catalog and user instructions for the system I checked above

Name \_\_\_\_\_

Title \_\_\_\_\_

Institution/Company \_\_\_\_\_

Department \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Phone ( ) \_\_\_\_\_

U.M.I. Article Clearinghouse

Mail to: University Microfilms International

300 North Zeeb Road, Box 99, Ann Arbor, MI 48106

# Advanced 1541 Disk Commands

Dave Straub

*If you want to go beyond the basics of Commodore disk programming, you'll need to learn direct access disk commands. These powerful commands allow you to read and write individual blocks on a disk. However, since improper use can irretrievably scramble a disk, they are recommended for intermediate and advanced programmers only. You should experiment with them on a scratch disk before attempting to manipulate any important files. The techniques work on any Commodore computer with a 1541 disk drive.*

The 1541 disk drive is a complex device, often called an *intelligent peripheral* because it contains its own microprocessor and operating system. With most computers (such as Apple, Atari, and IBM), the Disk Operating System (DOS) is a program you must load into the computer before using the disk drive. Commodore's DOS, on the other hand, is permanently stored in Read Only Memory (ROM) inside the drive itself.

The Commodore system has some real advantages: DOS does not take up any of the computer's memory, and it's available the instant you turn on the drive. It also makes the 1541 drive independently programmable. By sending *direct access* commands to the drive, you can read or write to any area on the disk, read or write to the drive's internal memory, and even run your own ML programs in the drive.

In this article we'll cover three commands used to manipulate individual disk blocks. A block, also called a *sector*, is a small area on the

disk that stores 256 bytes of data. As shown in your *1541 User's Manual*, each disk is divided into 35 separate tracks, with each track subdivided into anywhere from 17 to 21 individual blocks or sectors. This yields a total of 683 blocks, each with its own track and block number. For example, the first part of the disk directory is stored in track 18, block 1.

## The Command Channel

The first step in most Commodore disk programs is to open the command channel to the drive. This is a special channel used to send instructions to the drive and check for errors. Open it with a line similar to this:

```
10 OPEN 3,8,15
```

This opens communications on channel number 3 to device number 8 (the drive) with a secondary address of 15. The *channel* (sometimes called a *logical file*) number can be anything from 1-15. Once a channel is open, GET#, INPUT#, and PRINT# statements are used with the *channel number* to send or retrieve information on that channel. The number following the # character in these statements must match the channel number used in the OPEN statement. For example, if the channel is opened with OPEN 3,8,15, then PRINT#3,"10" sends an initialization command to the drive on the command channel.

The *device number* specifies which drive is accessed. The device number of the 1541 drive is always 8 unless you change it through software or by modifying the drive.

A *secondary address* of 15 has a

special meaning: It activates the command channel regardless of what channel number is used. For example, OPEN 1,8,15 or OPEN 15,8,15 both activate the command channel, using channel numbers 1 and 15, respectively.

In this article the command channel is used only to send direct access commands to the drive. However, the command channel also serves the important function of relaying drive error messages to the computer, as explained in your *1541 User's Manual*.

## Buffer Channels

The command channel's abilities are vital but limited: It can only transmit commands and error messages. To transfer *data* (information stored on the disk), you must open a second channel. When this is done, the drive sets aside a 256-byte *buffer* area within its internal memory. It's no coincidence that the buffer is exactly the right size for storing a block of disk data. Since all data moves through the drive's buffers, this type of channel is often called a *buffer channel*, although data channel might be a more descriptive term. To open a buffer channel, use "#" as a filename in an OPEN command:

```
OPEN 2,8,2,"#"
```

This statement tells the 1541 to open buffer channel 2 to device number 8 with a secondary address of 2, and the special filename "#" reserves a buffer in the drive. The secondary address can be any number from 2-14.

Now that the channel is open, you can find out which buffer the 1541 has reserved. Use GET# to

retrieve the first character available from that channel:

```
10 OPEN 2,2,"#":GET#2,X$
20 PRINT "BUFFER"ASC(X$+CHR$
    (0))"USED"
30 CLOSE 2
```

This program opens a data channel and retrieves the number of the buffer reserved for that channel. The 1541 has five 256-byte buffers located at these addresses:

```
Buffer 0 $0300-03FF (786-1023)
Buffer 1 $0400-04FF (1024-1279)
Buffer 2 $0500-05FF (1280-1535)
Buffer 3 $0600-06FF (1536-1791)
Buffer 4 $0700-07FF (1792-2047)
```

In most cases you needn't worry about which buffer is reserved for your data. The 1541 manages the buffers by itself and always reserves one for you unless none is available. However, by adding a number after the # character, you can force the drive to reserve a specific buffer. For instance, the statement OPEN 2,2,"#1" makes the drive set aside buffer number 1 for channel number 2.

To avoid needless errors, don't specify a buffer unless you really need to do so. The buffers are also the 1541's main data area, and at any given time one or more of them may already be in use. For instance, buffer 4 stores the Block Availability Map (BAM) of the current disk and is almost never available. The 1541 generates a NO CHANNEL error message when you try to use a buffer that's already reserved or try to access a channel that wasn't properly opened.

## Block-Read

The Block-Read command does exactly what the name implies, reading a block of information from the disk and storing it in a data buffer in the drive. Once the block has been read, you can transfer all or part of it to the computer's memory with GET# or INPUT# statements.

Block-Read has two alternate forms, one that works as expected and another that doesn't. Despite what your 1541 *User's Guide* says, don't use the B-R form of Block-Read. Use the alternate form (U1) instead. U1 always reads an entire block in correct order, beginning with the first byte of the block and ending with the last. To see how U1 works, type in and save Program 1

below. Program 1 works as listed on the Commodore 64 and the 128 in 64 mode. For the unexpanded VIC, change line 0 to:

```
0 POKE 36879,30:X1=7680:X2=384
88 rem 212
```

For the Commodore 16 or Plus/4, ignore the .rem numbers at the end of each line (they are for the VIC/64 "Automatic Proofreader" program) and change line 0 to:

```
0 X1=3872:X2=2848:COLOR8,2
```

Before running Program 1, put an unimportant disk in the drive. Direct access commands are very powerful; even a slight typing error in these programs can garble an entire disk, destroying all of its data. Until you have gained some experience with these commands, it's best to practice on a disk that doesn't contain important programs or data.

When you run Program 1, it displays the 256 bytes stored in track 18, block 0 of your disk. Among other things, this disk block contains the disk name and ID. Line 10 of the program opens the command channel (to send commands to the drive) and line 20 opens the buffer channel to reserve a data buffer. Line 30 contains the Block-Read command (U1). Note that line 30 uses PRINT#3 to send the U1 command. Block-Read is an instruction to the drive; like other instructions it must be sent via the command channel (in this case channel 3).

The actual command is enclosed in quotation marks. First comes the command itself, followed by several parameters separated by spaces. The first number after U1 is 2, telling the drive to read the block into the buffer reserved for channel 2.

The next parameter must always be 0 to indicate drive number 0. (This is a holdover from the old Commodore PET dual drives that are numbered 0 and 1. The 1541 drive is always drive 0, even when two 1541s with different device numbers are daisy-chained together.) The last two parameters inside quotes are the track and sector numbers of the block you wish to read. In this case, 18 and 0 are used

to read block 0 of track 18 from the disk.

In line 50 of Program 1, GET#2 retrieves data from the buffer channel. (Since GET# reads incoming zero bytes as null characters—which would crash the ASC function with an error message—it's always necessary to concatenate X\$ with CHR\$(0).) Line 70 ends the program by closing both channels. Since channel 2 was the last channel opened, it is the first one closed. Always end a program of this type by closing every channel you opened.

## Buffer-Pointer

U1 always reads a whole block, but in some cases you'll be interested in only part of the block. For example, you might want one program name from a directory block. The Buffer-Pointer command (abbreviated B-P) is designed for just such occasions. It points the drive to a designated byte within the data buffer, allowing you to read only the bytes you want. The general procedure is to read a block into the buffer with U1, set the pointer with B-P, then retrieve the desired bytes with GET#.

Type in and save Program 2, then run it. The program first reads the block from track 18, block 1 into a buffer. Then B-P sets the buffer pointer to byte 5. Like other direct access commands, B-P is also enclosed in quotes. The second parameter in the command is 2, telling the drive to use buffer channel 2. After 2 comes 5, the number of the byte you want to point at. The remainder of the program simply prints the name of the first program stored on the disk.

## Block-Write

The Block-Write command is the opposite of Block-Read, letting you write a block of data from a buffer to any block on the disk. Block-Write also comes in two forms, good and bad. Skip the B-W command; it has the same defects as B-R. Use U2 whenever you want to write a block of data to disk.

To see Block-Write at work, first load Program 2, then add the lines listed below as Program 3. Note that line 80 of Program 3 replaces line 80 of Program 2, and be sure to save a copy of this program

before you run it. This program combines all three of the commands discussed so far. First it reads the name of the first program in the disk directory from block 1 of track 18, then it asks you to type in a new program name. If the new name is less than 16 characters, line 110 "pads" the end of the name with SHIFtEd space characters.

You're almost ready to write the new name back to the data buffer. But first it's necessary to set the buffer pointer back to the spot where the old name begins in the buffer (line 120). This step is critical, because GET# affects the buffer pointer much as PRINT affects the cursor on the screen. Every time GET# fetches a character from the buffer, the pointer automatically moves one byte forward. Lines 50-70 repeated GET# 16 times, moving the pointer 16 bytes forward. Thus, before writing the new name back into the buffer, it's necessary to reset the pointer with a second B-P command.

Line 130 writes the new name to the buffer; since the name is data (not an instruction), it's transmitted over the data channel with PRINT#2. Line 140 sends the U2

command over channel 3 to complete the process, telling the drive to write the entire contents of the buffer back to track 18, block 1 of the disk.

As you've probably gathered by now, the 1541 drive handles disk data in block-sized chunks. Though you may want to change only one or two bytes in a block, it's necessary to read the whole block into the drive, make the changes, then write the altered block back to disk again. There's much more to direct access programming, of course, but you can do a great deal with these few commands, moving freely around the disk to examine or modify whatever you find.

Please refer to "COMPUTE's Guide to Typing in Programs" in this issue before entering the following listings.

### Program 1: Block-Read Demo

```

8 POKE53201,1:X1=1824:X2=55296      :rem 146
5 PRINTCHR$(147)CHR$(9)CHR$(14)      :rem 15
  :FORJ=1TO18:PRINT:NEXT              :rem 15
10 OPEN3,0,15                          :rem 188
20 OPEN2,0,2,"*"                       :rem 27
30 PRINT#3,"U1 2 8 18 0"             :rem 55
40 FOR X=0 TO 255                      :rem 79

```

```

50 GET#2,X$:Y=ASC(X$+CHR$(0))         :rem 99
60 POKEX1+X,Y:POKEX2+X,0:NEXT         :rem 113
70 CLOSE2:CLOSE3                      :rem 242

```

### Program 2: Buffer-Pointer Demo

```

10 OPEN3,0,15                          :rem 188
20 OPEN2,0,2,"*"                       :rem 27
30 PRINT#3,"U1 2 8 18 1"              :rem 56
40 PRINT#3,"B-P 2 5"                   :rem 221
50 FORK=0TO15:GET#2,X$:               :rem 49
60 IFX$<>CHR$(160)THENPRINTX$+CHR$(0): :rem 244
70 NEXT                                :rem 166
80 CLOSE2:CLOSE3                      :rem 243

```

### Program 3: Block-Write Demo

```

80 PRINT:PRINT"WHAT IS THE NEW NAME?" :rem 251
90 INPUT C$                             :rem 96
100 FORJ=1TO16:D$=D$+CHR$(160)         :rem 14
  :NEXT                                  :rem 99
110 C$=LEFT$(C$+D$,16)                 :rem 12
120 PRINT#3,"B-P 2 5"                   :rem 9
130 PRINT#2,C$                          :rem 9
140 PRINT#3,"U2 2 8 18 1"              :rem 187
200 CLOSE2:CLOSE3                      :rem 29

```

©

# IBM Filecopy

John Klein and Jeff Klein

*Here's a fast and easy way to backup multiple files on your disks for safe-keeping. It works on any IBM PC, PCjr, or compatible with at least 64K RAM and one or two floppy disk drives.*

What is rule number one when you use a computer? Always make backup copies of all important files.

But despite one of the most powerful disk operating systems in personal computing, that rule isn't always easy to follow on the IBM. DOS's DISKCOPY utility indiscriminately copies the entire disk, while the COPY command backs up only individual files. Neither allows you to copy groups of specific files from disk to disk or directory to directory very easily. Even if you

have two drives, it's time-consuming to combine files from several disks onto a single backup disk, or to backup a group of updated files. As a result, many of us don't make backups as often as we should.

"IBM Filecopy" offers a solution to this problem. It's a utility program that works on any IBM PC, PCjr, or true compatible with one or two disk drives. Filecopy lets

Table 1: Using IBM Filecopy

Type of copy wanted:	Target path to enter:
Same drive, directory "TEST"	\TEST (DOS 2.1 only)
Drive B, same directory	B:
Drive B, directory "TEST"	B:\TEST

Table 2: End of Copy Phase

Screen Message:	Operation/Response:
REM *** COPY COMPLETE	Copies completed.
A>ERASE FILECOPY.BAT	Batch file erases itself.
Insert disk with batch file and press any key when ready	DOS error message. Press any key
A>^C	Press BREAK.
Terminate batch job (Y/N)?	DOS message. Type Y.
A>	Backup process finished; returned to DOS.

you backup disks, directory by directory, or selectively backup individual files. These files can be of any type: BASIC, binary, command, etc. The files can be copied to any subdirectory on any other disk or the same disk.

### Using Filecopy

Filecopy is a BASIC program which creates a temporary DOS batch file to copy the specified files to the backup destination. When run, the program first asks you to insert the source disk. This is the disk which contains the files you want to backup. Then the program asks for the source directory of the source disk. If you're not copying from a subdirectory, just type N at this prompt. In either case, Filecopy reads the filenames from the source directory and stores them in an array for later use.

Next the program asks you for the *target path*, the destination for the backups. Type B: for drive B or A: for drive A (be sure to type the colon after the drive designator—B: instead of B). Then type *[directory name]* if you're copying the files to a subdirectory on the destination disk. You don't have to specify a directory if you're copying the files to the root (default) directory. If you're using a one-drive system, type B: for the target path as if you really have two drives. Never specify the same drive and directory as the source drive and directory, because the program won't copy files onto themselves. (See Table 1 for sample copy procedures.)

Filecopy then displays each filename from the source directory and asks if you want a copy. Simply type Y for each file you want copied, or N for those you don't want

copied. Note: When Filecopy encounters a subdirectory name on your source disk, it's fooled into thinking the subdirectory is a regular file. Since it can't copy subdirectory names, you must answer N when the program asks if you want to backup the subdirectory.

After Filecopy queries you on all of the filenames, it asks for confirmation: "Is this all okay?" If you accidentally typed a wrong Y or N at a previous prompt, type N at this one to get another chance. When you confirm your choices by typing Y, Filecopy stores the names of the files you want copied in a temporary batch file on the source disk. (If the source disk is write-protected, an error message appears and the program halts.) Then it returns you to DOS.

The next step is to type FILECOPY at the DOS prompt. This commands the batch file to copy each of the files you selected from the source disk to the target disk and directory. If you're using a one-drive system, DOS asks you to swap disks as it copies each file.

After the backup is complete, the temporary batch file erases itself off the source disk. This brings up an unavoidable DOS error message, "Insert disk with batch file and press any key when ready." When this message appears, just press CTRL-BREAK on the PC or Fn-B on the PCjr. Another DOS message asks if you want to terminate the batch job. Answer yes. The backup process is now complete. (Table 2 shows the screen messages and responses that should appear during this phase.)

### Additional Tips

If you have another BASIC program in memory before running

Filecopy, remember to save it on disk. Otherwise it will be replaced when you load and run Filecopy.

Filecopy works with all versions of DOS, but subdirectories are supported only in DOS 2.1 or higher. Do not specify directory changes in the target or source paths if you're using an earlier version of DOS.

If you have two drives and generally use the first drive for the DOS disk and the second for your programming disk, change the first statement in line 100 from P\$="A:" to P\$="B:". This lets you keep the source disk in drive B and put the target disk in drive A.

### IBM Filecopy

Please refer to "COMPUTE's Guide to Typing in Programs" before entering this listing.

```

10 SCREEN 8:WIDTH 80:COLOR 7,
  8
20 CLS:PRINT "Insert source disk in drive A and hit any key to continue"
30 GOSUB 700:A$=INPUT$(1)
40 FILES
50 PRINT:INPUT "Enter directory change (N=None) > ",DIR$
60 IF DIR$="N" OR DIR$="n" THEN
  EN 100
70 ON ERROR GOTO 720
80 CHOIR DIR$
90 ON ERROR GOTO 8:GOTO 40
100 P$="A:" :GOSUB 420:CLS
110 '*** get target path ***
120 INPUT "target path> ",PAT$
130 IF PATH$="B:" OR PATH$="b" OR PATH$="A:" OR PATH$="a" OR PATH$="" THEN 19
140 PRINT:INPUT "Do you want to create this directory on the target disk (Y/N) ",A$ IF A$="N" OR A$="n" THEN 190 ELSE IF A$="Y" AND A$="y" THEN 140
150 ON ERROR GOTO 720
160 MKDIR PATH$:PRINT:P$
170 '*** get files to copy ***
180 ON ERROR GOTO 8

```

```

G 190 FOR Z=0 TO FILENUM
E 200 PRINT "COPY> ";FILE$(Z);
TAB(24);"?";COLOR 31,0
H 210 GOSUB 700
H 220 AS=INPUT$(1);IF AS<"Y" A
ND AS<"N" AND AS<"N" AN
Q AS<"N" THEN 220
LC 230 PRINT SPC(21);AS;COLOR 7,0
G 240 IF AS="Y" OR AS="N" THEN
TY(Z)=1 ELSE TY(Z)=0
H 250 NEXT Z
E 260 INPUT "Is this all okay (
Y/N) ";AS
G 270 IF AS="N" OR AS="n" THEN
CLS;GOTO 190
H 280 IF AS<"Y" AND AS<"y" TH
EN 260
LC 290 "*** batch file creation
***
H 300 OPEN "FILECOPY.BAT" FOR O
UTPUT AS #1
G 310 PRINT #1,"VERIFY ON"
G 320 FOR Z=0 TO FILENUM
H 330 IF TY(Z)=0 THEN 350
G 340 PRINT #1,"COPY ";FILE$(Z)
";"PRTHS
H 350 NEXT Z
G 360 PRINT #1,"REM *** COPY CO
MPLTE"
H 370 PRINT #1,"ERASE FILECOPY.
BAT"
H 380 CLOSE #1

```

```

H 390 "*** enter DOS ***
H 400 CLS:SYSTEM
H 410 "*** directory read in ar
ray FILE$ ***
K 420 DEF SEG=0
K 430 CLS;COLOR 31,0;PRINT"One
moment please"
H 440 COLOR 7,0;ON ERROR GOTO 4
00
H 450 FILES P%:ON ERROR GOTO 0;
GOTO 470
H 460 BEEP;COLOR 31;CLS;PRINT"C
annot read directory";COL
OR 7;ON ERROR GOTO 0;END
G 470 DIM TEM$(40);LOCATE 3,1;C
OLOR 7;ROWS=0
H 480 POKE 1050,30;POKE 1052,34
;POKE 1054,0;POKE 1055,79
;POKE 1056,13;POKE 1057,2
0
E 490 LINE INPUT TEM$(ROWS)
H 500 IF TEM$(ROWS)<"*" THEN RO
WS=ROWS+1;GOTO 480
E 510 DIM FILES$(ROWS*4-1),TY(RO
WS*4-1)
E 520 ROWS=ROWS-1
G 530 FOR Z=0 TO ROWS
E 540 FOR Z1=0 TO 3
G 550 TS=MID$(TEM$(Z),Z1*4+1,1
7)
G 560 IF TS<"*" THEN FILES$(FILE
NUM)=FILES$(FILENUM)+1

```

```

H 570 NEXT Z1,Z
H 580 ERASE TEM$:FILENUM=FILENU
M-1
H 590 DEF SEG
H 600 "*** remove spaces from f
ilename ***
H 610 FOR Z=0 TO FILENUM
G 620 AS="";PERIOD=0
G 630 FOR Z1=1 TO 17
H 640 IF MID$(FILES$(Z),Z1,1)=""
THEN 660 ELSE IF MID$(F
ILES$(Z),Z1,1)="" THEN PE
RIOD=1
H 650 AS=AS+MID$(FILES$(Z),Z1,1)
G 660 NEXT Z1
G 670 FILES$(Z)=AS;IF PERIOD=0 T
HEN FILES$(Z)=FILES$(Z)+","
E 680 NEXT Z;RETURN
H 690 "*** clear keyboard buffe
r ***
H 700 DEF SEG=0;POKE 1050,PEEK(
1052);RETURN
H 710 "*** error messages ***
H 720 BEEP;COLOR 31,0;PRINT "O;
rectory does not exist";C
OLOR 7,0;RESUME 50
H 730 BEEP;COLOR 31,0;PRINT;PRI
NT;PRINT "Cannot create d
irectory -- reenter path"
;COLOR 7,0;RESUME 120

```

# Apple Text Windows

Daniel L. Joynt

*If you use an Apple II-series computer, you may have cast longing eyes at the flashy windowing capabilities of the Macintosh. What you may not realize is that the Apple II has a built-in windowing feature of its own—the text window. The following techniques work on any Apple II-series computer with at least 48K RAM.*

While it's difficult for an Apple II-series computer to emulate the slick graphics of the Macintosh, the Apple II does have a simple windowing capability known as the *text window*. Basically, a text window is a section on a low- or high-resolution graphics screen where text can be printed. The text window is easily controlled, too.

You're probably already famil-

iar with some aspects of the Apple II text window. When the computer is in text mode, in effect the text window covers the entire screen—you can print characters wherever you like. In graphics modes, the text window is confined to four lines at the bottom of the screen. When the four-line window fills up, text scrolls off the top of the window to make room for new text at the bottom. Ordinarily the computer manages the text window automatically. However, with a few simple POKEs you can manipulate the text window on your own—controlling its size and screen location.

To see a demonstration of four different text windows, type in, save, and run Program 1. To add windowing to your own BASIC

programs, you can use the subroutine in Program 2 (see instructions below).

## Open Your Own Windows

As you know, the Apple text screen is 40 columns wide and 24 rows tall. Any screen location can be defined in terms of an X (column) coordinate and a Y (row) coordinate. X coordinates range from 0 (far left) to 39 (far right). Y coordinates range from 0 (top) to 23 (bottom).

The Apple text window is controlled with four memory locations at decimal addresses 32, 33, 34, and 35. By POKEing values in these locations, you can set the boundaries of the window anywhere on the screen. Location 32 defines the left boundary of the text window; it

takes any value from 0-39, corresponding to X coordinate values. Location 33 defines the maximum length of text lines inside the window, which effectively sets the right boundary; it accepts any value from 1-40.

Memory location 34 defines the top boundary of the text window. It takes any value from 0-23, corresponding to Y coordinates. Location 35 defines the window's bottom boundary; it accepts values from 1-24, where 24 is the bottom row of the screen.

To keep a program from crashing, you must keep all the boundaries on the screen. For instance, the values in locations 32 and 33 when added together must not exceed 40. Otherwise, the right boundary would be off the screen. Illogical configurations—such as setting the top boundary below the bottom boundary—also cause a crash.

## Give The Cursor A Home

Once the window is opened, you'll want to fill it with text. Unfortunately, resetting locations 32-35 does not automatically relocate the cursor inside the new window. The HOME command sends the cursor to the upper-left corner, but also erases everything inside the window. To move the cursor less destructively, you can use the HTAB and VTAB commands.

HTAB and VTAB set the screen location where all following text will be printed. HTAB moves the text pointer to a specified column (1-40), and VTAB moves the text pointer to the designated row (1-24). Note that HTAB and VTAB act differently when used outside of a text window.

## An Easy Window Routine

Program 2 is a subroutine that creates a text window of any size and shape and even surrounds it with a border if you like. It uses line numbers 10000-10070, but you can renumber the lines when adding the routine to your own programs. Note that this is a subroutine, not a complete program; certain steps must be followed when using the routine after it has been added to another program.

Before calling the subroutine with GOSUB 10000, you must give each of the following variables a value within the range indicated. In addition to setting these variables, your program should not use the variables WX and WY, which the routine uses to draw borders.

### Variable Range

WL	2-39
WR	2-39
WT	2-22
WB	2-22
WS	any single character
WI	0-2

Line 10000 resets the text window to the size of the entire screen. This insures that HTAB and VTAB place the border properly. In lines 10010-10030 the variable WI defines the print mode of the border. A value of 0 maintains NORMAL mode, 1 sets the mode to INVERSE, and 2 sets it to FLASH.

The variable WS defines the character used for the border, which is drawn in lines 10040-10050. If you don't want a border, define WS as a null string (WS=""). Line 10060 locates the new text window, setting the left, right, top, and bottom boundaries with the variables WL, WR, WT, and WB, respectively. Note that the left and right boundaries (WL and WR) must be in the range 2-39 to allow room for a border. For the same reason, WT and WB (top and bottom boundaries) should be in the range 2-22. Once the window is defined, line 10070 resets the Apple to NORMAL mode, clears the inside of the new text window, and ends the routine.

## Program 1: Window Demonstration

```

2# 100 TEXT
4# 110 HOME
F# 120 REM --- DRAW MENU WINDOW
F# 130 WL = 10:WR = 30:WT = 5:WB
    = 10:WS = "X":WI = 0
4# 140 GOSUB 10000
4# 150 PRINT
F# 160 PRINT TAB( 4 ); "==== MENU
    =====
4# 170 PRINT : PRINT
4# 180 FOR Y = 1 TO 4
4# 190 PRINT TAB( 4 ); Y; " ~ WINDOW
    (" CHRS (64 + Y); " ) "
4# 200 NEXT Y
4# 210 PRINT : PRINT : PRINT
4# 220 PRINT "CHOOSE WINDOW (1-4
    ) : "
4# 230 GET Y$
4# 240 IF Y$ < "1" OR Y$ > "4" T
    HEN GOTO 100

```

```

4# 250 W = VAL (Y$)
4# 260 ON W GOSUB 1000, 2000, 3000,
    4000
4# 270 FOR Z = 1 TO 1000: NEXT Z
4# 280 LIST
4# 290 HOME
4# 300 PRINT : PRINT : PRINT
4# 310 PRINT TAB( 3 ); "WINDOW (" ;
    CHR$ (64 + W); " ) "
4# 320 FOR Z = 1 TO 1000: NEXT Z
4# 330 REM --- DRAW MENU WINDOW
4# 340 WL = 6:WR = 34:WT = 12:WB
    = 12:WS = CHR$ (32):WI =
    1
4# 350 GOSUB 10000
4# 360 PRINT TAB( 5 ); "PLEASE PRE
    SS ANY KEY"
4# 370 GET Y$
4# 380 GOTO 130
4# 1000 REM --- DRAW WINDOW (A)
F# 1010 WL = 2:WR = 15:WT = 2:WB
    = 3:WS = CHR$ (32):WI =
    1
4# 1020 GOSUB 10000
4# 1030 RETURN
4# 2000 REM --- DRAW WINDOW (B)
4# 2010 WL = 26:WR = 39:WT = 2:WB
    = 8:WS = "X":WI = 1
4# 2020 GOSUB 10000
4# 2030 RETURN
4# 3000 REM --- DRAW WINDOW (C)
4# 3010 WL = 2:WR = 15:WT = 16:WB
    = 22:WS = "X":WI = 0
4# 3020 GOSUB 10000
4# 3030 RETURN
F# 4000 REM --- DRAW WINDOW (D)
4# 4010 WL = 26:WR = 39:WT = 16:WB
    = 22:WS = "X":WI = 1
4# 4020 GOSUB 10000
4# 4030 RETURN
4# 10000 POKE 32,0: POKE 33,40:
    POKE 34,0: POKE 35,24
4# 10010 IF WI = 0 THEN NORMAL
4# 10020 IF WI = 1 THEN INVERSE
4# 10030 IF WI = 2 THEN FLASH
4# 10040 FOR WY = WT - 1 TO WB +
    1: PRINT WY; HTAB WL +
    1: PRINT WY; HTAB WR +
    1: PRINT WY; NEXT
4# 10050 FOR WX = WL TO WR: VTAB
    WT - 1: HTAB WX: PRINT
    WY; VTAB WB + 1: HTAB
    WX: PRINT WY; NEXT
4# 10060 POKE 32,WL - 1: POKE 33,
    WR - WL + 1: POKE 34,W
    T - 1: POKE 35,WB
4# 10070 NORMAL : HOME : RETURN

```

## Program 2: Window Subroutine

```

4# 10000 POKE 32,0: POKE 33,40:
    POKE 34,0: POKE 35,24
4# 10010 IF WI = 0 THEN NORMAL
4# 10020 IF WI = 1 THEN INVERSE
4# 10030 IF WI = 2 THEN FLASH
4# 10040 FOR WY = WT - 1 TO WB +
    1: PRINT WY; HTAB WL -
    1: PRINT WY; HTAB WR +
    1: PRINT WY; NEXT
4# 10050 FOR WX = WL TO WR: VTAB
    WT - 1: HTAB WX: PRINT
    WY; VTAB WB + 1: HTAB
    WX: PRINT WY; NEXT
4# 10060 POKE 32,WL - 1: POKE 33,
    WR - WL + 1: POKE 34,W
    T - 1: POKE 35,WB
4# 10070 NORMAL : HOME : RETURN

```

# COMPUTE!'s Guide To Typing In Programs

Before typing in any program, you should familiarize yourself with your computer. Learn how to use the keyboard to type in and correct BASIC programs. Read your manuals to understand how to save and load BASIC programs to and from your disk drive or cassette unit. Computers are precise—take special care to type the program exactly as listed, including any necessary punctuation and symbols, except for special characters as noted below. To help you with this task, we have implemented a special listing convention as well as a program to help check your typing—the "Automatic Proofreader." Please read the following notes before typing in any programs from COMPUTE!. They can save you a lot of time and trouble.

Commodore, Apple, and Atari programs can contain some hard-to-read (and hard-to-type) special characters, so we have developed a listing system that indicates the function of these control characters. (There are no special control characters in our IBM or TI-99/4A listings.) You will find Commodore and Atari special characters within curly braces; do not type the braces. For example, {CLEAR} or {CLR} instructs you to insert the symbol which clears the screen on the Atari or Commodore machines. For Commodore, Apple, and Atari, a symbol by itself within curly braces is usually a control key or graphics key. If you see {A}, hold down the CTRL key and press A. This will produce a reverse video character on the Commodore (in quote mode), a graphics character on the Atari, and an invisible control character on the Apple. Commodore computers also have a special control key labeled with the Commodore logo. Graphics characters entered with the Commodore logo key are enclosed in a special bracket that looks like this: {<A>}. In this case, you would hold down the Commodore logo key as you type A. Our Commodore listings are in uppercase, so shifted symbols are underlined. A graphics heart symbol (SHIFT-S) would be listed as S. One exception is (SHIFT-SPACE). When you see this, hold down SHIFT and press the space bar. If a number precedes a symbol, such as {5 RIGHT}, {6

S}, or {<8 Q>}, you would enter five cursor rights, six shifted S's, or eight Commodore-Q's. On the Atari, inverse characters (printed in white on black) should be entered after pressing the inverse video key.

Since spacing is sometimes important, any more than two spaces will be

listed. For example, {6 SPACES} means to press the space bar six times. Our listings never leave a space at the end of a line, instead moving it to the next printed line as {SPACE}. For your convenience, we have prepared this quick-reference chart for the Commodore and Atari special characters:

## Atari 400/800/XL/XE

When you see	Type	See
{CLEAR}	ESC SHIFT <	W Clear Screen
{UP}	ESC CTRL -	↑ Cursor Up
{DOWN}	ESC CTRL =	↓ Cursor Down
{LEFT}	ESC CTRL +	← Cursor Left
{RIGHT}	ESC CTRL +	→ Cursor Right
{BACK S}	ESC DELETE	␣ Backspace
{DELETE}	ESC CTRL DELETE	⌫ Delete character
{INSERT}	ESC CTRL INSERT	⌫ Insert character
{DEL LINE}	ESC SHIFT DELETE	⌫ Delete line
{INS LINE}	ESC SHIFT INSERT	⌫ Insert line
{TAB}	ESC TAB	→ TAB key
{CLR TAB}	ESC CTRL TAB	→ Clear tab
{SET TAB}	ESC SHIFT TAB	⌫ Set tab stop
{BELL}	ESC CTRL 2	🔔 Ring buzzer
{ESC}	ESC ESC	⌨ ESCape key

## Commodore PET/CBM/VIC 64/128/16/+4

When You Read:	Press:	See:	When You Read:	Press:	See:
{CLR}	SHIFT CLR/HOME		{ 1 }		
{HOME}	CLR/HOME		{ 2 }		COMMODORE
{UP}	SHIFT ↑ C/SK ↓		{ 3 }		COMMODORE
{DOWN}	↓ C/SK ↓		{ 4 }		COMMODORE
{LEFT}	SHIFT ← C/SK →		{ 5 }		COMMODORE
{RIGHT}	→ C/SK →		{ 6 }		COMMODORE
{RVS}	CTRL 9		{ 7 }		COMMODORE
{OFF}	CTRL 0		{ 8 }		COMMODORE
{BLK}	CTRL 1		{ P1 }		
{WHT}	CTRL 2		{ P2 }	SHIFT	
{RED}	CTRL 3		{ P3 }		
{CYN}	CTRL 4		{ P4 }	SHIFT	
{PUR}	CTRL 5		{ P5 }		
{GRN}	CTRL 6		{ P6 }	SHIFT	
{BLU}	CTRL 7		{ P7 }		
{YEL}	CTRL 8		{ P8 }	SHIFT	
				←	



## The Automatic Proofreader

We have developed a series of simple, yet effective programs that can help check your typing. Type in the appropriate Proofreader program listed below, then save it for future use. On the VIC, 64, or Atari, run the Proofreader to activate it, then enter NEW to erase the BASIC loader (the Proofreader remains active, hidden in memory, as a machine language program). Pressing RUN/STOP-RESTORE or SYSTEM RESET deactivates the Proofreader. You can use SYS 886 to reactivate the VIC/64 Proofreader, or PRINT USR(1536) to reactivate the Atari Proofreader. On the Apple, the Proofreader automatically erases the BASIC portion of itself after you activate it by typing RUN, leaving only the machine language portion in memory. It works with either DOS 3.3 or ProDOS. Disable the Apple Proofreader by pressing CTRL-RESET before running another BASIC program. The IBM Proofreader is a BASIC program that simulates the IBM BASIC line editor, letting you enter, edit, list, save, and load programs that you type. Type RUN to activate.

Once the Proofreader is active, try typing in a line. As soon as you press RETURN, either a decimal number (on the Commodore), a hexadecimal number (on the Apple), or a pair of letters (on the Atari or IBM) appears. The number or pair of letters is called a *checksum*. Try making a change in the line, and notice how the checksum changes.

All you need to do is compare the value provided by the Proofreader with the checksum printed in the program listing in the magazine. In Commodore listings, the checksum is a number from 0 to 255. It is set off from the rest of the line with *rem*. This prevents a syntax error if the checksum is typed in, but the REM statements and checksums need not be typed in. It is just there for your information.

In Atari, Apple, and IBM listings, the checksum is given to the left of each line number. Just type in the program one line at a time (without the printed checksum) and compare the checksum generated by the Proofreader to the checksum in the listing. If they match, go on to the next line. If not, check your typing: You've made a mistake. On the Commodore, Atari, and Apple Proofreaders, spaces are not counted as part of the checksum, so be sure you type the right number of spaces between quote marks. The Commodore and Atari Proofreaders do not check to see that you've typed the characters in the right order, so if characters are transposed, the checksum still matches the listing. Because of the checksum meth-

od used, do not type abbreviations, such as ? for PRINT. The IBM Proofreader is the pickiest of all; it will detect errors in spacing and transposition. Be sure to leave Caps Lock on, except when typing lowercase characters.

## IBM Proofreader Commands

Since the IBM Proofreader replaces the computer's normal BASIC line editor, it has to include many of the direct-mode IBM BASIC commands. The syntax is identical to IBM BASIC. Commands simulated are LIST, LLIST, NEW, FILES, SAVE, and LOAD. When listing your program, press any key (except Ctrl-Break) to stop the listing. If you type NEW, the Proofreader prompts you to press Y to be sure you mean yes.

Two new commands are BASIC and CHECK. BASIC exits the Proofreader back to IBM BASIC, leaving the Proofreader in memory. CHECK works just like LIST, but shows the checksums along with the listing. After you have typed in a program, save it to disk. Then exit the Proofreader with the BASIC command, and load the program in BASIC as usual (this replaces the Proofreader in memory). You can now run the program, but you may want to re-save it to disk. The version of your program that you re-save from BASIC will take up less space on disk and will load faster, but it can no longer be edited with the Proofreader. If you want to convert a program to Proofreader format, save it to disk with SAVE "filename".A.

## Special Proofreader Notes For Commodore Cassette Users

The Proofreader resides in a section of memory called the cassette buffer, which is used during tape LOADs and SAVEs. Therefore, be sure to press RUN/STOP-RESTORE to get the Proofreader out of the way before saving or loading a program. If you want to use the Proofreader with tape, run the Proofreader, then enter these two lines exactly as shown, pressing RETURN after each one:

```
AS="PROOFREADER.T":BS="(10  
SPACES)"FOR X=1 TO 4:AS=AS  
+BS:NEXT  
FOR X=886 TO 1018:AS=AS+CHR$(  
PEEK(X)):NEXT-OPEN 1,1,AS:  
CLOSE1
```

Then insert a blank tape and press RECORD and PLAY to save a special version of the Proofreader. Anytime you need to reload the Proofreader after it has been erased—for example, after you reload a partially completed program—just rewind the tape, type OPEN1:CLOSE1, then press PLAY.

You'll see the message FOUND PROOFREADER.T, but not the familiar LOADING message. Don't worry; the Proofreader is in memory. When READY comes back, enter SYS 886.

## Program 1: VIC/64 Proofreader

By Charles Brannon, Program Editor

```
10 PRINT"[CLR]PLEASE WAIT...":  
FORI=886TO1018:READA:CK=CK+  
A:POKEI,A:NEXT  
20 IF CK<17539 THEN PRINT"  
[DOWN]YOU MADE AN ERROR"[PR  
INT]"IN DATA STATEMENTS."*EN  
D  
30 SYS886:PRINT"[CLR][2 DOWN]P  
ROOFREADER ACTIVATED."*NEW  
40 DATA 173,836,003,201,158,20  
8,001,096,141,151,003,173  
50 DATA 037,003,141,152,003,16  
9,150,141,036,003,169,003  
60 DATA 141,037,003,169,000,13  
3,254,096,032,007,241,133  
70 DATA 251,134,252,132,253,00  
8,201,013,240,017,201,032  
80 DATA 240,005,024,101,254,13  
3,254,165,251,166,252,164  
90 DATA 253,040,096,169,013,03  
2,210,255,165,214,141,251  
100 DATA 003,206,251,003,169,0  
00,133,216,169,019,032,210  
110 DATA 255,169,018,032,210,2  
55,169,58,032,210,255,166  
120 DATA 254,169,000,133,254,1  
72,151,003,192,007,200,006  
130 DATA 032,205,109,076,235,0  
03,032,205,221,169,032,032  
140 DATA 210,255,032,210,255,1  
73,251,003,133,214,076,173  
150 DATA 003
```

## Program 2: Atari Proofreader

By Charles Brannon, Program Editor

```
100 GRAPHIC 0  
110 FOR I=1536 TO 1700:RE  
AD A:POKE I,A:CK=CK+A  
:NEXT I  
120 IF CK<>19072 THEN ? "   
Error in Data Stateme  
nts. Check Typing."*  
END  
130 A=USR(1536)  
140 ? I ? "Automatic Proof  
reader Now Activated."  
  
150 END  
160 DATA 104,100,0,185,26  
3,201,09,240,7  
170 DATA 200,200,192,34,2  
00,243,96,200,169,74  
180 DATA 153,26,3,200,169  
6,153,26,3,162  
190 DATA 0,189,0,220,157,  
74,6,232,224,16  
200 DATA 200,245,169,93,1  
41,70,6,169,6,141  
210 DATA 79,6,24,173,4,22  
0,105,1,141,95
```

```

220 DATA 6,173,5,220,105,
    8,141,96,6,169
230 DATA 8,133,203,96,247
    ,239,125,241,93,6
240 DATA 244,241,115,241,
    124,241,76,205,238
250 DATA 8,8,8,8,32,62,
    246,8,201
260 DATA 155,240,13,201,3
    2,240,7,72,24,101
270 DATA 203,133,203,104,
    40,96,72,152,72,130
280 DATA 72,160,0,169,128
    ,145,88,200,192,40
290 DATA 200,249,165,203,
    74,74,74,74,24,105
300 DATA 161,160,3,145,88
    ,165,203,41,15,24
310 DATA 105,161,200,145,
    88,169,0,133,203,104
320 DATA 170,104,168,104,
    40,96

```

### Program 3: IBM Proofreader

By Charles Brannon, Program Editor

```

10 "Automatic Proofreader Ver
    sion 2.00 (Lines 270,510,5
    15,517,620,630 changed from
    v.1)
100 DIM L$(500),LNUM$(500):COL
    OR 0,7,7:KEY OFF:CLS:MAX=
    0:LNUM(0)=45536!
110 DN ERR0R GOTO 120:IF KEY 15,
    CHR$(4)+CHR$(70):DN KEY(1
    5) GOSUB 640:KEY (15) DN:
    GOTO 130
120 RESUME 130
130 DEF SEG=M440:W=PEEK (M444)
140 ON ERR0R GOTO 650:PRINT:P
    RINT"Proofreader Ready."
150 LINE INPUT L$:V=CSRLIN-IN
    T(LEN(L$)/W)-1:LOCATE Y,1
160 DEF SEG=0:POKE 1050,30:PO
    KE 1052,34:POKE 1054,0:PO
    KE 1055,79:POKE 1056,13:P
    OKE 1057,28:LINE INPUT L$
    :DEF SEG:IF L$="" THEN 15
    0
170 IF LEFT$(L$,1)="" THEN L
    $=MID$(L$,2):GOTO 170
180 IF VAL(LEFT$(L$,2))=0 AND
    MID$(L$,3,1)="" THEN L$
    =MID$(L$,4)
190 LNUM=VAL(L$):TEXTS=MID$(L
    $,LEN(STR(LNUM))+1)
200 IF ASC(L$)>57 THEN 260 "n
    o line number, therefore
    command
210 IF TEXT="" THEN GOSUB 540
    0:IF LNUM=LNUM(P) THEN GO
    SUB 560:GOTO 150 ELSE 150
220 CSUM=0:FOR I=1 TO LEN(L$
    ):CSUM=(CSUM+ASC(MID$(L
    $,I,1)))+1 AND 255:NEXT:LOC
    ATE Y,1:PRINT CHR$(65+CSUM
    UM/16)+CHR$(65+CSUM AND
    15)+" "
230 GOSUB 540:IF LNUM(P)=LNUM
    THEN L$(P)=TEXT:GOTO 15
    0 'replace line
240 GOSUB 580:GOTO 150 'inser
    t the line
260 TEXT="" :FOR I=1 TO LEN(L
    $):A=ASC(MID$(L$,I)):TEXT
    $=TEXT$+CHR$(A+32$(A>96 A
    ND A<123)):NEXT

```

```

270 DELIMITER=INSTR(TEXT$," "
    ):COMMAND$=TEXT$:ARG$=""
    :IF DELIMITER THEN COMMAN
    $=LEFT$(TEXT$,DELIMITER-1
    ):ARG$=MID$(TEXT$,DELI
    METER+1) ELSE DELIMITER=IN
    STR(TEXT$,CHR$(34)):IF DELI
    METER THEN COMMAND$=LEFT$
    (TEXT$,DELIMITER-1):ARG$=
    MID$(TEXT$,DELIMITER)
280 IF COMMAND$<>"LIST" THEN
    410
290 OPEN "scrns" FOR OUTPUT A
    S #1
300 IF ARG$="" THEN FIRST=0:P
    =MAX-1:GOTO 340
310 DELIMITER=INSTR(ARG$,"-")
    :IF DELIMITER=0 THEN LNUM
    =VAL(ARG$):GOSUB 540:FIR
    ST=1:GOTO 340
320 FIRST=VAL(LEFT$(ARG$,DELI
    METER)):LAST=VAL(MID$(ARG
    $,DELIMITER+1))
330 LNUM=FIRST:GOSUB 540:FIR
    ST=LNUM:LAST=GOSUB 540:IF
    F=0 THEN P=MAX-1
340 FOR X=FIRST TO P:P=N:MID$(
    STR(LNUM(X)),2)+" "
350 IF CKFLAG=0 THEN AS="" :GO
    TO 370
360 CSUM=0:AS=N$+L$(X):FOR I
    =1 TO LEN(AS):CSUM=(CSUM
    +ASC(MID$(AS,I,1)))+1 AND
    255:NEXT:AS=CHR$(65+CSUM
    /16)+CHR$(65+CSUM AND 1
    5)+" "
370 PRINT #1,AS+N$+L$(X)
380 IF INKEY$<>" " THEN X=P
390 NEXT X:CLOSE #1:CKFLAG=0
400 GOTO 130
410 IF COMMAND$="LLIST" THEN
    OPEN "lpt1:" FOR OUTPUT A
    S #1:GOTO 300
420 IF COMMAND$="CHECK" THEN
    CKFLAG=1:GOTO 290
430 IF COMMAND$<>"SAVE" THEN
    450
440 GOSUB 600:OPEN ARG$ FOR O
    UTPUT AS #1:ARG$="" :GOTO
    300
450 IF COMMAND$<>"LOAD" THEN
    490
460 GOSUB 600:OPEN ARG$ FOR I
    NPUT AS #1:MAX=0:P=0
470 WHILE NOT EOF(1):LINE INP
    UT #1:L$=LNUM(P)+VAL(L$):
    L$(P)=MID$(L$,LEN(STR(VA
    L(L$)))+1):P=P+1:NEXT
480 MAX=P:CLOSE #1:GOTO 130
490 IF COMMAND$="NEW" THEN IN
    PUT "Erase program - Are
    you sure":L$:IF LEFT$(L$,
    1)="" THEN 130:ELSE 130
500 IF COMMAND$="BASIC" THEN
    COLOR 7,0,0:ON ERR0R GOTO
    0:CLS:END
510 IF COMMAND$<>"FILES" THEN
    520
515 IF ARG$="" THEN ARG$="A:"
    ELSE SEL=1:GOSUB 600
517 FILES ARG$:GOTO 130
520 PRINT"Syntax error":GOTO
    130

```

```

540 P=0:WHILE LNUM<LNUM(P) AN
    D P<MAX:P=P+1:NEXT:RETURN
560 MAX=MAX+1:FOR X=P TO MAX:
    LNUM(X)=LNUM(X+1):L$(X)=
    L$(X+1):NEXT X:RETURN
580 MAX=MAX+1:FOR X=MAX TO P+
    1 STEP -1:LNUM(X)=LNUM(X-
    1):L$(X)=L$(X-1):NEXT X:L$
    (P)=TEXT$:LNUM(P)=LNUM:RET
    URN
600 IF LEFT$(ARG$,1)<>CHR$(34
    ) THEN 520 ELSE ARG$=MID$(
    ARG$,2)
610 IF RIGHT$(ARG$,1)=CHR$(34
    ) THEN ARG$=LEFT$(ARG$,LE
    N(ARG$)-1)
620 IF SEL=0 AND INSTR(ARG$,
    ".")=0 THEN ARG$=ARG$+"BA
    S"
630 SEL=0:RETURN
640 CLOSE #1:CKFLAG=0:PRINT"S
    topped."
650 PRINT "Error #":ERR:RESUM
    E 150

```

### Program 4: Apple Proofreader

By Tim Victor, Editorial Programmer

```

10 C = 0: FOR I = 76B TO 76B +
    6B: READ A: C = C + A: POKE I
    ,A: NEXT
20 IF C < > 7250 THEN PRINT "ER
    ROR IN PROOFREADER DATA ST
    ATEMENTS": END
30 IF PEEK (190 * 256) < > 76 T
    HEN POKE 56,0: POKE 57,3: CA
    LL 1002: GOTO 50
40 PRINT CHR$(4):"INNA300"
50 POKE 34,0: HOME : POKE 34,1:
    VTAB 2: PRINT "PROOFREADER
    INSTALLED"
60 NEW
100 DATA 216,32,27,253,201,141
110 DATA 200,60,138,72,169,0
120 DATA 72,189,255,1,201,160
130 DATA 240,8,104,10,120,255
140 DATA 1,105,0,72,202,200
150 DATA 230,164,170,41,15,9
160 DATA 40,201,50,144,2,233
170 DATA 57,141,1,4,130,74
180 DATA 74,74,74,41,15,9
190 DATA 40,201,50,144,2,233
200 DATA 57,141,0,4,104,170
210 DATA 169,141,96

```



# from COMPUTER HOUSE Specialties...

## American Covers

Leather Look - Durable  
for all Drive Computers &  
Monitors - CALL

## Axiom Interfaces

Parallax - AT	69.95
Parallax - II	69.95
Parallax - CD	69.95

## Dishwasher Spike Masters

FP-1000	59.95
P-1000	49.95
P-500	18.95

## Carli Battery Pack Caring Case

199.95

## MPP Modems

1000E	89.95
1064	79.95
Video 7 C-64 Modem	39.95

## MPP

Interface 1150	63.95
Print Buffer	119.95
1200 Baud MPP Modem	Call

## Indus Disk Drives

GT-Atari	259.95
GT-C-64	269.95
GT-Apple	179.95

## Intronic Kameleon

Apple 2C M-100 Interface	69.95
AS Switch Box w/cables	99.00

## Jasco Printer Cable

IBM Printer Cable	24.95
-------------------	-------

## MTR Monitor

IBM-TTL Green or Amber	169.95
w/switch & tilt base	

## Star Micronics, Inc.

SG10	239.95
SG10C	259.00
SD10	Call
SD15	Call
SR10	Call
SR15	Call
SR18	Call

## COMPUTER HOUSE Specialties

P.O. Box 26

Belton, MO 64012

Call Toll Free to Order

Outside Kansas call

1-800-255-0390 then 454457

In Kansas Call

1-800-332-0274 then 454457

We accept checks, money order, Visa & MasterCard (include # & expiration date). Please add 3% for credit card purchases. Personal & Company checks allow 2 weeks for clear. School orders accepted. Please subject to change without notice. Dealer inquires invited. Missouri residents add 6% sales tax. Visa MasterCard, Company Checks or Money Orders. Orders will be shipped next day. Due to low prices of some items final returns must have original authorization number. Call to obtain one.

**COMPUTER  
HOUSE  
Specialties**

```

460 GETAS:IFA$>"*THENPRINT:PRI
NT:GOTO310      :rem 25
470 NEKPK:PRINTCHR$(20);:NEXTI
:PRINT:PRINT:GOTO310
:rem 50
480 IFN<0 THEN PRINT:GOTO310
:rem 168
490 A(J)=N:NEXTJ      :rem 199
500 CKSUM=AD-INT(AD/256)+256:F
ORI=1T06:CKSUM=(CKSUM+A(I)
)AND255:NEXT      :rem 208
510 PRINTCHR$(10);:GOSUB70:PR
INTCHR$(146);      :rem 94
511 IFN=1THENA=6:GOTO315
:rem 254
515 PRINTCHR$(20);:IFN=CKSUMTHE
N530      :rem 122
520 PRINT:PRINT"LINE ENTERED W
RONG : RE-ENTER":PRINT:GOS
UB1000:GOTO310      :rem 176
530 GOSUB280      :rem 218
540 FORT=1T06:POKEAD+I-1,A(I):
NEXT:POKE54272,0:POKE54273
,0      :rem 227
550 AD=AD+IF AD<E THEN 310
:rem 212
560 GOTO 710      :rem 188
570 N=0:2=0      :rem 88
580 PRINT"$$$";      :rem 81
581 GETAS:IFA$>"*THEN581
:rem 95
582 AV=- (AS="M")*2*(AS="")-3*(
AS=".")-4*(AS="J")-5*(AS="
K")-6*(AS="L")      :rem 41
583 AV=AV-7*(AS="U")-8*(AS="I"
)-9*(AS="O"):IFA$="H"THENA
S="0"      :rem 134
584 IFAV=0THENA$=CHR$(48+AV)
:rem 134
585 PRINTCHR$(20);:A=ASC(A$):I
FA=13ORA=44ORA=32THEN670
:rem 229
590 IFA>128THENM=A:RETURN
:rem 137
600 IFA<20 THEN 630      :rem 18
610 GOSUB690:IFI=LANDT=44THENM
=1:PRINT"[OFF]"[LEFT]
[LEFT]::GOTO690      :rem 62
620 GOTO570      :rem 109
630 IFA<40ORA>57THEN580
:rem 105
640 PRINTAS:I=N*M+10*A=40
:rem 106
650 IFN=255 THEN A=20:GOSUB100
:GOTO600      :rem 229
660 Z=Z+1:IFZ<3THEN580      :rem 71
670 IFZ=0THENGOSUB1000:GOTO570
:rem 114
680 PRINT",":RETURN      :rem 240
690 SV=PEEK(289)+256*PEEK(210)
+PEEK(211)      :rem 149
691 FORT=1T03:T=PEEK(SV-I)
:rem 67
695 IPT<44ANDT<58THENPOKESV-
1,32:NEXT      :rem 205
700 PRINTLEFT$("{3 LEFT}",I-1)
:RETURN      :rem 7
710 PRINT"[CLR]"[RVS]*** SAVE *
**[3 DOWN]      :rem 236
715 PRINT"[2 DOWN]"[PRESS [RVS]
RETURN[OFF] ALONE TO CANCE
L SAVE]"[DOWN]      :rem 106
720 F$="":INPUT"[DOWN] FILENAM
E":F$:IFF$=""*THENPRINT:PRI
NT:GOTO310
730 PRINT:PRINT"[2 DOWN]"[RVS]T
[OFF]APE OR [RVS]D[OFF]ISK
(T/D)"      :rem 228
740 GETAS:IFA$>"*T*ANDA$>"D"T
HENT740      :rem 36

```

```

750 DV=1-7*(AS="D"):IFDV=8THEN
F$="0":+F$:OPEN15,8,15,"S"
+F$:CLOSE15      :rem 212
760 TS=F$:ZK=PEEK(53)+256*PEEK
(54)-LEN(T$):POKE782,ZK/25
6      :rem 3
762 POKE781,ZK-PEEK(782)+256:P
OKE780,LEN(T$):SYS65469
:rem 189
763 POKE780,1:POKE781,DV:POKE7
82,1:SYS65466      :rem 69
765 K=S:POKE254,K/256:POKE253,
K-PEEK(254)+256:POKE780,25
3      :rem 17
766 K=K+1:POKE782,K/256:POKE78
1,K-PEEK(782)+256:SYS65496
:rem 235
770 IF(PEEK(783)AND1)OR(191AND
ST)THEN780      :rem 111
775 PRINT"[DOWN]"[DONE]"[DOWN]"
:GOTO310      :rem 113
780 PRINT"[DOWN]"[ERROR ON SAVE.
[2 SPACES]"TRY AGAIN."*IFDV
=1THEN720      :rem 171
781 OPEN15,8,15:INPUT#15,E1$E
2$:PRINTI$;E2$:CLOSE15:GO
T0720      :rem 103
790 PRINT"[CLR]"[RVS]*** LOAD *
**[2 DOWN]"      :rem 212
795 PRINT"[2 DOWN]"[PRESS [RVS]
RETURN[OFF] ALONE TO CANCE
L LOAD)"      :rem 82
800 F$="":INPUT"[2 DOWN] FILEN
AME":F$:IFF$=""*THENPRINT:G
OTO310      :rem 144
810 PRINT:PRINT"[2 DOWN]"[RVS]T
[OFF]APE OR [RVS]D[OFF]ISK
(T/D)"      :rem 227
820 GETAS:IFA$>"*T*ANDA$>"D"T
HENT820      :rem 14
830 DV=1-7*(AS="D"):IFDV=8THEN
F$="0":+F$:OPEN15,8,15,"S"
+F$:CLOSE15      :rem 157
840 TS=F$:ZK=PEEK(53)+256*PEEK
(54)-LEN(T$):POKE782,ZK/25
6      :rem 2
841 POKE781,ZK-PEEK(782)+256:P
OKE780,LEN(T$):SYS65469
:rem 187
845 POKE780,1:POKE781,DV:POKE7
82,1:SYS65466      :rem 70
850 POKE780,0:SYS65493      :rem 11
860 IF(PEEK(783)AND1)OR(191AND
ST)THEN870      :rem 111
865 PRINT"[DOWN]"[DONE."*GOTO310
:rem 96
870 PRINT"[DOWN]"[ERROR ON LOAD.
[2 SPACES]"TRY AGAIN."[DOWN]
*IFDV=1THEN800      :rem 172
880 OPEN15,8,15:INPUT#15,E1$E
2$:PRINTI$;E2$:CLOSE15:GO
T0800      :rem 102
1000 REM BUZZER      :rem 135
1001 POKES4296,15:POKES4277,45
:POKES4278,165      :rem 207
1002 POKES4276,33:POKE54273,6
:POKES4272,5      :rem 42
1003 FORT=1T0200:NEXT:POKE5427
6,32:POKE54273,0:POKE5427
2,0:RETURN      :rem 282
2000 REM BELL SOUND      :rem 78
2001 POKES4296,15:POKES4277,0
:POKES4278,247      :rem 152
2002 POKE 54276,17:POKE54273,4
0:POKE54272,0      :rem 86
2003 FORT=1T0100:NEXT:POKE5427
6,16:RETURN      :rem 57
3000 PRINTCS;"[RVS]NOT ZERO PA
GE OR ROM":GOTO1000
:rem 89

```

## COMMODORE 64

Bloodroot-D	27.95	Shanty Snow-Cart	34.95
Broomed-0	27.95	Blue Max 2001-D	28.95
Comp. Quarterback-D	27.95	Flint System 8-D	49.95
Computer Ambush-D	45.95	Forever Wood-D	34.95
Kampgrape-D	41.95	White Now-Cart	34.95
Dyer Marker Garden-D	34.95	Brace Lite-D	27.95
Gemstone Marker-D	23.95	Conan-D	27.95
Impetuous Galactico-D	27.95	Fast Load-Cart	25.95
Phantom-D	27.95	Net Worth-D	49.95





ATARI	
800 XL Computer	CALL
1050 Drive	169
1350 Recorder	64
1350	52
1350	52
1027	239
650	109

BRODERBUND	
Bank St. Writer	42.75
The Print Shop	29.95
Simon's Star	29.95
Speller	19.95
Speller	19.95

BUSINESS	
VISCALC	\$159.75
LETTER PERFECT II	59.00
DATA PERFECT	\$69.75
FILE MANAGER	\$69.75
HOME FILE MGR	\$69.75

ADVENTURE	
Dishy	29.95
Ultra Disassembler	29.95

GRAPHIC TABLET	
Digital	39.95
Koala Pad	39.95

EASTERN HOUSE	
Monkey Wrench 1800	24.95
Monkey Wrench 1800 XL	24.95

CONTINENTAL	
Home Accountant	44.95
1985 Book of Allen Software	15.95

ADVENTURE INTERNATIONAL	
Digital	39.95
Ultra Disassembler	39.95

WICO	
(Joysticks)	
15-9714 Bat Handle	16.99
50-2022 Super 3-way	19.99
72-4545 Trackball	29.99

HAYES	
Smartmodem 300	169
Smartmodem 1200	450
Smartmodem 1200B	375
Micromodem III	249
Micromodem III	249
Chronograph	175
Smart Com II	75

TELE LEARNING	
CM-250 GC-64	29.95
AP-250 (Apple)	109.95
16-250 (IBM)	109.95

Scarborough	
Songwriter	\$24.75
Picturement	\$24.75
Phi Beta F	\$32.75
Masterpiece	\$24.75
Run I Money	\$32.75
Net Worth	\$54.75

Microprose	
Solo Flight	\$22.75
Question	\$22.75
Spelth	\$22.95
P-15 Strike	\$22.75
Air Rescue	\$22.75

SSI	
Baseball	\$22.75
Field of Fire	\$22.75
Germany 1985	\$22.75
50 Missions	\$21.75

Spinnaker	
Alphabet	\$18.75
Story Machine	\$19.75
Kids on Keys	\$19.75
Grandma	\$19.75
Snooper Trap	\$22.75

Broderbund	
Bank St. Writer	\$42.75
Bank St. Filer	\$42.75
Bank St. Master	\$42.75
Bank St. Spell	\$42.75
Bank of Sun	\$24.95
Print Shop	32.95
Lode Runner	\$22.95

Graphics Tablet	
Spelketch	\$49.95
Koala	\$64.95

COMMODORE	
Simon's Basic	34.75
Assembler 64	34.75
Super Expander	29.75
Logo 64	49.75
Logo 64	49.75
Easy Calc	34.75
Easy Script	36.75
C-64 Computer	CALL
C-164 Disk Drive	175
MPS 801 Printer	175
C-163 Monitor	39.75
C-163 Joystick	39.75
C-1660 Auto Modem	85
C-1660 Auto Modem	85

SS1	
(C-64)	
Computer Baseball	24.75
Field of Fire	24.75
Computer Quarterback	24.75
Question	24.75
50 Missions Crush	24.75

Scarborough	
Songwriter	\$24.75
Phi Beta F	\$32.95
Masterpiece	\$24.75
Run I Money	\$34.75
Net Worth	\$49.95

SUBLOGIC	
Flight Simulator II	22.75
Night Mission Pinball	22.75

PERSONAL PERIPHERALS	
Super Sketch 64	32.75
Printer Utility	10.75

KOALA	
(C-64)	
Koala Pad	59.95

NOVATION	
J-Cal	69
Smart Cal Plus	319
Smart Cal 103	319
Smart Cal 103/212	369
Auto Cal	269
Auto Cal	269
Auto Cal II	239
212 Apple Cal	339
Apple Cal 212	249

CARCOC MOD-1 (C-64)	
CALL	
NEC TRUCK (C-64)	CALL
MITEE MO (C-64)	CALL
1600 AUTO MODEM (C-64)	85
COMPUERVE	29.95

CARDCO	
G01 Wire Now	29.95
G02 Wire Now - 64	39.95
G01 Mail Now - 64	29.00
G04 Spell Now - 64	29.00
G02 Utility Desk	19.95
CSD - Disk Drive (new)	CALL
ACD - Modem (new)	CALL
G03 Tax Paper (new)	27.95
G07 Calc Now (new)	27.95
D08 Super Printer Utility	
D08 Super Printer	
Utility	27.95
CK-1 Numeric Keypad	34.95
CB/5 5 Slot	39.95
Board C-64	54.00
CR/1 Light Pen	29.75
CE/1 Cassette Interface	29.75
CB/3 3 Slot	
Board Vio-20	24.95
CB/6 6 Slot	
Board Vio-80	65.00

HES	
HES Games 64	29.95
Omni Writer Spell	34.95
HES Mon 64	23.95
Microplot Multipen	55.00
Type N Write	19.95
Turtle Graphics II	23.95
Cell Defense	22.95
Paint Brush	12.95
Hi Math	22.95
Graphics Basic	27.95
HES K1	29.95
Milchone	23.95
64 Fonth	24.95
HES Winter 64	24.95

BATTERIES INCLUDED	
Paper Clip	59.95
Spell Pak	34.95
Consultant	59.95
Paper Clip	
WSPR Pak	75.95
Home Pak	34.95
Bus Card	129.95
80 Column Board	109.95

Microprose	
Solo Flight	\$22.75
NATO	\$22.75
Spelth	\$19.95
P-15 Strike	\$22.75
Air Rescue	\$22.75

★ SUPER SPECIAL! ★

★ LOWEST PRICES! ★

## MONITORS

AMDEK	
320 Green	125
320 Amber	139
320 Amber IBM	135
Color 300 Audio	245
Color 520 Composite	369
Color 620	429
Color 700	495
Color 710	569

PANASONIC	
DT 1350 RGB composite	329

SAKATA	
SC-100 Color	219
ST-151 Stand	29
SC-1000 Green	99
SA 1030 Amber	109

PRINCEXON GRAPHICS	
MAX-12 Amber	169
HX-12 RGB	475
SR-12 RGB	599

NEC	
JB-1260 Green	95
JB-1201 Green	135
JB-1215 Color	275
JB-1216 RGB	325
JB-1480 Color	365
JB-1205 Amber	139

GORILLA	
12" Green	74
8" Amber	79

ZENITH	
ZVM 1225A Amber	64
ZVM 1235 Green	79
ZVM 124 Amber IBM	129
ZVM 131 Color	275
ZVM 135 RGB	389
ZVM 135 Composite	449
ZVM 135 Hi Res Color	589

TAXAN	
210 Color RGB	249
115 Green	119
115 Amber	132
Color RGB	132
410 Color RGB	339
420 Color IBM	429
115 Green IBM	129
122 Amber IBM	145

X-TRON	
Colorator I Composite Green	199

AMERICA'S MAIL ORDER HEADQUARTERS  
LYCO COMPUTER  
WORLD'S LEADER IN SALES & SERVICE

TO ORDER  
CALL TOLL FREE  
800-233-8760  
In PA 1 717-327-1824  
Lyco Computer  
P.O. Box 5088  
Jersey Shore, PA 17740

# PRINTER INTERFACES

Cardco G+ .....	\$69.95
Cardco 8 .....	\$49.95
Turbo Print .....	\$79.95

# MODEMS

Westridge Autodemod .....	\$69.95
Modem 300 (1680) .....	\$79.95
MPP 1064 C64 .....	\$79.95
Total Telecommunication .....	\$59.95

# PRINTERS

Commodore 801 .....	\$169.95
Commodore 803 .....	\$169.95
(tractor feed option) .....	\$ 29.95
Commodore DPS 1101 (Daisy Wheel) .....	\$299.95
Star SG-10 .....	\$224.95
Panasonic 1091 .....	\$299.95
Citizen MSP-10 .....	\$329.95
G.E. Printer .....	\$219.95

# MONITORS

1702 Color .....	\$199.95
Gorilla (green) .....	\$ 79.95
Gorilla (amber) .....	\$ 89.95
Sakata Color .....	\$179.95

# SOFTWARE

Word Processors	
Paper Clip (w/spell) .....	\$79.95
Word Pro 3 + (64) .....	\$39.95
Write Now .....	\$39.95
Data Bases	
Superbase 64 .....	\$59.95
The Consultant .....	\$79.95
Spread Sheets	
Practical 11 .....	\$49.95
Calc Result .....	\$69.95
Utility Software	
Print Shop .....	\$29.95
Floppicide Head Cleaner .....	\$14.95
Mach 5 .....	\$24.95

# COMPUTER WAREHOUSE

(In FL) 305-274-3680; 1-800-372-0214  
7222 S.W. 117th Avenue, Miami, FL 33183

Add 3% for shipping and handling charges. Add prices reflect 3% cash discount. Credit card orders should add 3%. Prices subject to change without notice.

# Professional Handicapping Systems



by Professor Jones

# OLD Thoroughbred "Gold" Edition™

Full featured handicapping system designed for the professional and serious bettor. \$159.95 complete

**OLD Enhanced "Gold" Edition™**  
"Gold" Edition with complete Master Rider™ system integrated into the same disk. This powerful program will analyze offerings and select better bets analysis with a "single keystroke." Master Rider™ accepted \$199.95 complete

# OLD Limited "Gold" Edition™

Enhanced Professional Handicapping system designed to be the only system you need. It has important data program written based on a philosophy of risk and low risk of the handicapper's perspective. This program is designed to "save" the user the time and money in programming experience. (Contains integrated Master Rider™ \$299.95 complete)

# GG Gold Day Analysis™

The only professional day handicapping on the market includes:  
1) Stand 2) Post Today 3) Known 4) First Loss 5) Condition 6) Running Style 7) Weight 8) All day without weighting 9) 100 days indicator  
\$149.95 complete (with integrated Master Rider™ \$199.95)  
Limited Gold Day \$299.95

# PPE Professor Jones' Football Predictions, Print Pa™

Complete Football Analysis with Beta base.  
1) Overview 2) Super Plays 3) Date Base Stars 4) Point Spreads 5) Two-Unit Bet 6) Hired 100 Teams  
Highest percentage of winners 100%  
\$49.95 complete \$99.95 with On-Disk News Management

# MNH Master Horse Handicapping™

Professional software designed to provide a thorough analysis of all data and past data in North America and Canada.  
\$159.95 complete (with integrated Master Rider™ \$199.95)

# NDA Statistical™

The all base merged software will provide the user with all betting data from which winning winning information is taken.  
\$59.95 complete without \$149.95 with On-Disk Version

# LOT LOTTERY ANALYSIS™

Advanced computer program designed to search subtle patterns in weekly lottery numbers.  
\$19.95 complete without \$39.95 with code

# BROCHURE AVAILABLE

# IBM™

# APPLE™

# TRS-80™

# CPM™

# COMMODORE™

Prof. Jones  
1940 W. State St.  
Boise, ID 83702



48 HR. FREE SHIPPING

CALL  
208-342-6939

TECHNICAL SUPPORT  
SOFTWARE: Add \$5.00 load  
Add 1.99 per C.D. / UPS Blue  
\$5.00 / Out of Country \$9.00 /  
C.D. Residents: Add 13 weeks per  
\$5000 checks / Cash price only  
Add 2% Mkt. Mgr. / Prices sub  
ject to change.

# WHITE HOUSE COMPUTER

P.O. Box 4025

Williamsport, PA 17701

1-717-322-7700

HOURS: Monday thru Friday 9:00 am till 6:00 pm

VISA and MC Accepted 4%

PRINTERS		LEGEND		INTERFACES		DISK DRIVES		MONITORS	
PANASONIC						ATARI		ZENITH	
1090 .....	187.00	880 .....	209.00	MPP 1150 .....	52.00	Indus GT .....	215.00	122A .....	82.95
1091 .....	257.00	1080 .....	229.00	B50 (Atari) .....	108.00	1050 .....	165.95	123G .....	75.95
1092 .....	369.00	1380 .....	269.00	UPrint w/16K Buffer .....	79.95	"Happy" 1090 .....	365.95	TEKNIKA	
1093 .....	455.00	1385 .....	306.00	Cardco G .....	55.95	MSD Dual .....	459.00	M-10 Composite & Separate Video .....	
3151 .....	505.00	C-64		Cardco B .....	37.95	C-64		M-22 RGS Composite & Separate Video .....	
C. ITOH		MPS 801 .....	179.95	PRINTER PAPER		Indus GT .....	239.00	AMDEK	
8510A .....	285.00	802 .....	208.95	Lazer 500 sheets .....	11.95	MODEMS		XTRON	
8510BC1 .....	339.00	803 .....	175.95	Lazer 1000 sheets .....	15.95	C 1800 .....	44.95	300G .....	119.00
8510BC2 .....	389.00	MCS 801 .....	314.95	COMPUTERS		C 1850 .....	54.95	300A .....	129.00
8510BP1 .....	339.00	DPS 1101 .....	314.95	COMMODORE		C 1670 300/1200 .....	54.95	Color 300 .....	235.00
8510SP .....	389.00	ATARI		C 64 .....	149.95	Band .....	185.95	Color 500 .....	345.00
8510SR .....	429.00	KXM 201 Non-Impact Dot .....	99.95	C 128 .....	275.95	C 64 MPP 1064 .....	59.95	COMMODORE	
1550BCD .....	485.00	KTC 201 Color Non-Impact .....	109.95	ATARI		MPP 1000E .....	69.95	C-1700 .....	199.95
1550P .....	445.00	Matrix .....	109.95	800 XL .....	99.95	Telearning .....	39.95	C-1802 .....	204.95
A10-20P .....	465.00	XDM 121 Daisy Wheel .....	209.95	130 XT .....	145.95	Hayes 1200 .....	389.95	C-1902 RGB .....	259.95
F10-40PS .....	975.00	XMM 801 Dot Matrix .....	169.95	520 ST .....	CALL	Miley Mo .....	59.95	C-1901 Monochrome .....	129.95
F10-55PS .....	1059.00	Impact .....	169.95	SOFTWARE		Westridge .....	69.95	ATARI	
F10 Tractor .....	169.00	STC 504 Color .....	139.95	BRODERBUND "ATARI" & COMMODORE		Compuserve .....	21.95	Diskettes	
F10 Sheet Feeder .....	169.00	Non-Impact .....	139.95	Print Shop .....	29.95	Startar Kit .....	21.95	SKC	
STAR MICRONICS		STD 121 Daisy Wheel .....	219.95	Graphics Library .....	18.95	MAXWELL		ELEPHANT	
SG10 .....	217.00	SMM 801 Dot Matrix .....	279.95	BS Writer .....	41.95	MD-1 .....	15.95	MD-2 .....	21.95
SG15 .....	379.00	Impact .....	279.95	Strength .....	21.95	SS/SD .....	10.50	SS/SD .....	13.50
SD10 .....	345.00	CITIZEN		VIDEO CASSETTE RECORDER		OS/OD .....	15.50	POLICY	
SD15 .....	450.00	MSP-10 .....	325.00	\$295.95		MAXWELL		No deposit on COD orders	
SD18 .....	485.00	MSP-15 .....	479.00	Symphonic VHS/Remote		DISKETTES		FREE freight on all prepaid cash orders	
SR15 .....	585.00	MSP-20 .....	469.00	● 109 Channel Color Ready		SKC		over \$200 in the continental USA	
PowerType .....	309.00	MSP-25 .....	629.00	● 14 Day/1 Program Timer		MAXWELL		APD and PPD add \$5 per handover	
EPSON		OKIDATA		● 12 Pre-set Channels		DISKETTES		For priority mail add \$4 per handling	
RX80 .....	219.00	82 .....	219.95	● Auto Rewind		SKC		PA residents add \$4 per sales tax	
LX80 .....	245.00	84 .....	240.95	● 109 Channel Color Ready		SKC		Collective products must have Prior	
FX80 .....	389.00	86 .....	349.95	● 14 Day/1 Program Timer		SKC		PA number. Schools not 15.	
JX80 .....	529.00	90 .....	580.95	● 12 Pre-set Channels		SKC			
RX100 .....	359.00	CARDICO				SKC			
FX100 .....	490.00	LO1 .....	385.00			SKC			
		LO2 .....	208.00			SKC			



# What the world really needs is a 99 cent Double Sided, Double Density Diskette with a LIFETIME WARRANTY!

## And DISK WORLD! has it.

**Introducing Super Star Diskettes:  
the high quality diskette with  
the lowest price  
and the best LIFETIME WARRANTY!**

In the course of selling more than a million diskettes every month, we've learned something: higher prices don't necessarily mean higher quality.

In fact, we've found that a good diskette manufacturer simply manufactures a good diskette... no matter what they charge for it. (By way of example, consider that none of the brands that we carry has a return rate of greater than 1/1,000th of 1 percent.)

In other words, when people buy a more expensive diskette, they aren't necessarily buying higher quality. The extra money might be going toward flashier advertising, smazzier packaging or simply higher profits. But the extra money in a higher price isn't buying better quality.

All of the good manufacturers put out a good diskette Period.

### How to cut diskette prices ...without cutting quality.

Now this discovery posed a dilemma: how to cut the price of diskettes without lowering the quality.

There are about 50 companies claiming to be "diskette" manufacturers.

Trouble is, most of them aren't manufacturers. Rather, they are fabricators or marketers, taking other company's components, possibly doing one or more steps of the processing themselves and putting their labels on the finished product.

The new Eastman Kodak diskettes, for example, are one of these. So are IBM 5 1/4" diskettes. Same for DYSAN, Palsani and many, many other familiar diskette brand names. Each of these diskettes is manufactured in whole or in part by another company!

So, we decided to act just like the big guys. That's how we would cut diskette prices...without lowering the quality.

We would go out and find smaller companies to manufacture a diskette to our specifications... specifications which are higher than most... and simply create our own "name brand" diskette.

Name brand diskettes that offered high quality at low prices.

SUPER  
STAR  
DISKETTES  
★

5 1/4"  
SSDD  
.91 ea.  
Qty. 50

5 1/4"  
DSDD  
.99 ea.  
Qty. 50

Super Star diskettes are sold in multiples of 50 only. Diskettes are shipped with white Tynex sleeves, reinforced hubs, user ID labels and write protect tabs.

**Boy, did we get lucky. Our Super Star Diskettes are the same ones you've been using for years...without knowing it.**

In our search for the low priced, high quality diskette of our dreams, we found something even more interesting.

We found that there are several manufacturers who don't give a hoot about the consumer market for their diskettes. They don't spend millions of dollars in advertising trying to get you, the computer user, to use their diskettes.

Instead, they concentrate their efforts on turning out the highest quality diskettes they can...because they sell them to the software publishers, computer manufacturers and other folks who (in turn) put their name on them...and sell them for much higher prices to you!

All that, when a software publisher or computer manufacturer or diskette marketer puts their name on a diskette, they want it to work time after time, everytime. (Especially software publishers who have the nasty habit of copy-protecting their originals!)

**Super Star Diskettes. You already know how good they are. Now you can buy them...cheap.**

Well, that's the story.

Super Star diskettes don't roll off the beat from Page-Page or emerge from a basement plant just east of Nowhere.

Super Star diskettes have been around for years...and you've used them for years as copy-protected software originals, unprotected originals. Sometimes, depending on which computer you own, the system master may have seen an a Super Star diskette. And maybe more than once, you've bought a box or two or more of Super Star diskettes without knowing it. They just had some "big" company's name on them.

Super Star Diskettes are good. So good that a lot of major software publishers, computer manufacturers and other diskette marketers buy them in the tens or hundreds of thousands.

We buy them in the millions.

And then we sell them to you.

Cheap.

**When every little bit counts,  
it's Super Star Diskettes.**

You've used them a hundred times...under different names.

Now, you can buy the real McCoy, the same diskettes that major software publishers, computer manufacturers and diskette marketers buy...and call them yours.

We simply charge less.

### Super Special!

Order 50 Super Star Diskettes and we'll be happy to sell you an Amway Modem-Mate 50 for only \$5.75, shipping included...a lot less than the suggested retail price of \$15.95.



Regular DISK WORLD price: \$10.95 ea.  
+ \$2.00 Shipng.

### The Super Star LIFETIME WARRANTY!

Super Star Diskettes are unconditionally warranted against defects in original material and workmanship so long as owned by the original purchaser. Returns are simple: just send the defective diskettes with proof of purchase, postage paid by you with a short explanation of the problem, and we'll send you the replacements. (Incidentally, coffee stained diskettes and diskettes with staples driven through them don't qualify as "defective".)

**WE WILL MEET OR BEAT ANY NATIONALLY  
ADVERTISED PRICE  
ON THE SAME PRODUCTS AND QUANTITIES  
SUBJECT TO THE SAME TERMS AND CONDITIONS.**

Suite 6806  
30 East Huron Street  
Chicago, Illinois 60611

### DISKETTE STORAGE CASES

#### PERFECTDATA DUAL "N FILE"

Perfectly holds 10 5 1/4" diskettes. Just flip the lever and they all slide up for easy access and identification. Gray with snaplock plastic front.

\$2.75 Ea. + .25 Shipng.

#### DISK CADDIES

The original flip-top holder for 10 5 1/4" diskettes. Beige or gray only.

\$1.65 Ea. + .20 Shipng.

#### DISKETTE 70 STORAGE

Out-free storage for 10 5 1/4" diskettes. 50 dividers included. In excellent style.

\$11.95 Ea. + \$3.00 Shipng.



### HOW TO ORDER:

#### ORDERS ONLY:

1-800-621-6627  
(In Illinois: 1-312-944-2760)

#### INQUIRIES:

1-312-944-2766

FOR FASTEST SERVICE, USE NO-COST MCI MAIL. Our address is DISK WORLD. It's a FREE MCI MAIL letter. No charge to you. (Satisfaction permitting, we'll ship these orders in 24 hours or less.)

**SHIPPING:** 5 1/4" & 5 1/4" DISKETTES—Add \$3.00 per each 100 or fewer diskettes. **OTHER ITEMS:** Add shipping charges as shown in addition to other shipping charges. **PAIDMENT:** VISA, MASTERCARD and Prepaid orders accepted. **COD OR P.O.:** Add additional \$3.00 special handling charge. **A.P.O., F.P.O., A.I. & P.O. ORDERS:** Include shipping charges as shown and telephone 5% of total order amount to cover P.H. and insurance. We ship only to United States addresses, except for those listed above. **TAXES:** Illinois residents, add 9% sales tax.

**MINIMUM ORDER:** \$25.00 or 20 diskettes.

**HOURS:**  
Human: 9AM-6PM Central Time, Monday through Friday  
Answering Machine: 6PM-9AM, All Times  
MCI MAIL: 24 hours a day.

# DISK WORLD!, Inc.

# J&R MUSIC WORLD

23 PARK ROW, NEW YORK, N.Y. 10038

ORDER TOLL FREE 800-221-8180

WHEN BUYING MAIL ORDERS—WHO YOU BUY FROM IS AS IMPORTANT AS WHAT YOU BUY

RELIABLE

**THINSE GOLD STANDARD FLOPPY DISKS**

5.25" 5 1/4" 3.5" 3" 2 1/2" 2" 1 1/2" 1" 3/4" 3/8"

**\$49.95**

**TOP 10 COMPUTER HARDWARE**

AT&T 100% COMPATIBLE

**\$149.95**

**TOP 10 COMPLETE SYSTEMS**

COMPAQ 100% COMPATIBLE

**\$199.95**

**TOP 10 VIDEO EQUIPMENT**

SONY 100% COMPATIBLE

**\$149.95**

**TOP 10 AUDIO EQUIPMENT**

SONY 100% COMPATIBLE

**\$149.95**

**TOP 10 VIDEO EQUIPMENT**

SONY 100% COMPATIBLE

**\$149.95**

**TOP 10 VIDEO EQUIPMENT**

SONY 100% COMPATIBLE

**\$149.95**

**TOP 10 VIDEO EQUIPMENT**

SONY 100% COMPATIBLE

**\$149.95**

**TOP 10 VIDEO EQUIPMENT**

SONY 100% COMPATIBLE

**\$149.95**

**NOTE:**

ALL PRODUCTS CARRY U.S. WARRANTIES

**3 FREE CATALOGS**

23 PARK ROW, DEPT. CR, NYC, NY 10038

NAME: \_\_\_\_\_

ADDRESS: \_\_\_\_\_

CITY: \_\_\_\_\_ STATE: \_\_\_\_\_ ZIP: \_\_\_\_\_

# Advertisers Index

Reader Service Number/Advertiser	Page
102 Abacus Software	61
103 Abacus Software	64,65
104 American PEOPLE/LINK	21
Aprotek	2
Bartenes Included	9
105 Cordco, Inc.	IBC
106 Chase Manhattan	39
107 Comark, Inc.	27
Commodore	FC-1
108 CompuServe	5
Computability	91
109 Computer Direct	45
110 Computer House Specialties	90
111 Computer Mail Order	30,31
112 Computer Media Marketing	96
Computer Warehouse	94
Disk World, Inc.	39
Disk World, Inc.	95
First Class Peripherals	43
ICS Computer Training	61
Jason-Ranheim	32
113 Jensen Tools, Inc.	44

Reader Service Number/Advertiser	Page
114 J & R Music World	96
Lycia Computer	92,93
North Hills Corp.	96
North Hills Corp.	96
115 Pacific Exchanges	96
116 Prof. Jones	94
117 Protech	47
118 Startpoint Software	25
119 Startpoint Software	27
120 Strategic Simulations, Inc.	13
121 Strategic Simulations, Inc.	19
122 subLOGIC Corporation	11
123 Taxon	8C
124 Timeworks	32
125 White House Computer	94

Apple Titles from COMPUTE! Books	7
COMPUTE! Books August Releases	23
COMPUTE!'s Commodore Collection, Volumes 1 and 2	29
COMPUTE! Subscription	17

**3M Diskettes**

**Lifetime Warranty**

**TIRED OF WAITING FOR SERVICE AND PRICE?**

**9 out of 10 SURVEYED DISK BUYERS PREFERRED**

**NORTH HILLS**

**#1 IN SERVICE AND PRICE**

**1-800-328-3472**

Formatted and hard sectored disks in stock—Dealer inquiries invited. COD, VISA, MASTERCARD. All orders shipped within 24 hrs.

**NORTH HILLS CORP. INTERNATIONAL**

3564 Rolling View Dr  
White Bear Lake, MN 55110  
MN, call collect—612-770-0485

**MEMOREX FLEXIBLE DISCS**

**WE WILL NOT BE UNDER-SOLD!!** Call Free (800)235-4137 for prices and information. Dealer inquiries invited and COD's accepted.

**VISA**

**PACIFIC EXCHANGES**

100 Fourth Blvd  
San Luis Obispo, CA  
93401. In Cal call (800)592-5935 or (805)543-1037

**"No-Stretch" Viewing**

**ANGLED PRINTER STAND**

Standard & Deluxe Models available  
BD132 column available  
Deluxe fits over Disk Drive  
Priced from \$21.95

**COMPUTER MEDIA MARKETING**

Dept. [C-B] P.O. BOX 614  
Katy, TX 77452  
713/392-2385  
[collect ok]

**Dealer Inquiries Welcome**

**maxell DISKS**

**LIFETIME WARRANTY**

**TIRED OF WAITING FOR SERVICE AND PRICE?**

**9 out of 10 SURVEYED DISK BUYERS PREFERRED**

**NORTH HILLS**

**#1 IN SERVICE AND PRICE**

**1-800-328-3472**

Formatted and hard sectored disks in stock—Dealer inquiries invited. COD, VISA, MASTERCARD. All orders shipped within 24 hrs.

**NORTH HILLS CORP. INTERNATIONAL**

3564 Rolling View Dr  
White Bear Lake, MN 55110  
MN, call collect—612-770-0485

Technical

# Breakthrough

Number 27



## Price and performance no other printer interface can touch!

Only CARDCO would dare improve on its own best seller (the +G has far out-sold any other printer interface, and has set the industry standard by which competitors are judged). The G-Wiz is even better — and costs 30% less.\* Our 27th major innovation in Commodore accessories has all the +G's features, and more...

### **Built-In Buffer for More Speed**

The G-Wiz buffer dumps high-resolution screens up to 18 times faster than competitive interfaces without buffers. No more waiting! A complex 50-minute printout now takes just 2.5 minutes with the G-Wiz.

### **Exclusive! Aspect Ratio Selection**

Only the G-Wiz matches graphics characters to standard characters on Okidata, Epson, Star

Micronics, Prowriter, Smith Corona, C-ITOH, Gorilla Banana, and many other dot matrix printers. Now you can perfectly align high resolution graphics characters within text blocks, or in columns.

CARDCO excellence triumphs again! The G-Wiz is the "best bang for the buck" on the printer interface market today — and it's backed by CARDCO'S exclusive lifetime warranty! G-Wiz: another distant target for the competition to shoot at.

CARDCO, Inc. 300 S. Topeka/Wichita, KS 67202



\* Actual price may vary according to dealer and region.

**The Wizards from the Land of Oz Have Done It Again!**

# TAXAN

## We Make Your Apple Shine

At TAXAN we keep it simple. Our Apple line of Monitors combines many of the great features of our IBM line. Super High-Res and High-Res capabilities allow for the

Being #1 is an education. Picking the right Monitor for your Apple takes an educated guess, too! finest in business graphics. Video games look better, too! And your children will love you for choosing TAXAN instead of Brand X.



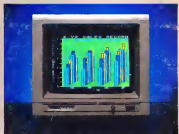
**MODEL 420** 12 inch Super High-Res RGB Color Monitor. Fully compatible with Apple, IBM and most other personal computers. 640x262 line resolution. Unlimited colors available through analog video circuit. Black face, non-glare tube.



**MODEL 220** 14" Composite Color Monitor with Audio. Standard NTSC composite video signal input. Chrominance and luminance inputs available for Commodore, Atari, and other computers. Extremely low geometric distortion. Switchable to green or amber display. Built-in 2 watt audio. **Optional:** Tilt/Swivel Base available (Model 110-12)



**410-08** RGB interface card for the Apple II, IIx, IIe and Franklin Ace 1000 and 1200 Rev. A only.  
**410-17** RGB Module for Apple IIc.  
**410-80B** 80 Column and RGB interface card. Compatible with Apple IIe. Third color cursor for text editing.  
**410-8064** Same as 410-80 with additional 64K Ram memory and 6 color double High-Res graphics. Text is color changeable.



**MODEL 410** 12 inch High-Res RGB Color Monitor. Fully compatible with Apple, IBM and most other personal computers. 512x382 line resolution. Unlimited colors available through analog video circuit.



**MODEL 116** 12 inch High-Res Amber Monitor. Long Persistence Phosphor with non-glass screen. Fully compatible with Apple, IBM and most other personal computers. 1000 line resolution. More than 20 MHz bandwidth. 2000 character display. Optional Tilt/Swivel Base available (Model 110-12). Green Phosphor available (Model 115).

**MODEL 500 PRINTER BUFFER**  
Centronics compatible parallel interface with 64K memory expandable to 256K. Four function modes include: Manual, Double, Free and Command. Compatible with most printers.

### OPTIONS—

**Model 599-01** 64K Extra Memory Add-on  
**Model 599-02** Extra parallel Cable



# TAXAN

So, get an education. See TAXAN today!  
**The Smart Choice.**

18005 Cortney Ct. City of Industry, CA 91748 (818) 810-1291

© 1985 TAXAN Corporation

\*Apple is a registered trademark of Apple Computers, Inc.

\*IBM is a registered trademark of International Business Machines, Inc.

# RETROMAGS

Our goal is to preserve classic video game magazines so that they are not lost permanently.

People interested in helping out in any capacity,  
please visit us at [www.retromags.com](http://www.retromags.com).

No profit is made from these scans, nor do we offer anything  
available from the publishers themselves.

If you come across anyone selling releases from  
this site, please do not support them and do let us know.

Thank you!

